

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

На правах рукопису

Кушніков Ігор Олександрович

Програмне забезпечення системи особистого онлайн органайзеру

Спеціальність: 123 «Комп'ютерна інженерія»

Освітній ступінь: бакалавр

Науковий керівник:

Улічев Олександр Сергійович

(підпис)

(дата)

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

_____ О.А. Смірнов

(підпис)

ПБ

« _____ » _____ 2021 р.

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
О.А.Смірнов
« 21 » грудня 2020 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Кушнікову Ігорю Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи особистого онлайн організатору

керівник роботи Улічев Олександр Сергійович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 189-02 від 28.12.2020 року

2. Строк подання студентом роботи до захисту 22.05.2021 р.

3. Мета та завдання кваліфікаційної бакалаврської роботи: Метою розробки є програмне забезпечення системи особистого онлайн організатору

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

6. Дата видачі завдання « 21 » грудня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.03.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2021 р.	
3.	Розробка моделі компонента	20.03.2021 р.	
4.	Розробка структур даних	25.03.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2021 р.	
6.	Програмування алгоритмів	10.04.2021 р.	
7.	Оформлення ПЗ	17.04.2021 р.	
8.	Попередній захист роботи	22.05.2021 р.	

Студент _____

(підпис)

_____ (прізвище та ініціали)

Керівник роботи _____

(підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Кушніков І.О. Програмне забезпечення системи особистого онлайн органайзеру. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

У даній кваліфікаційній бакалаврській роботі розроблено програмне забезпечення, яке призначено для системи особистого онлайн органайзеру.

Метою роботи є створення системи особистого онлайн органайзеру.

Результат роботи – програмна реалізація системи особистого онлайн органайзеру.

В процесі роботи над реалізацією системи виконано дослідження існуючих методів, алгоритмів та програмних засобів. Розроблено та реалізовано власне програмне забезпечення, здійснено опис всіх його компонентів.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10.

Програму розроблено на мові програмування PHP.

Ключові слова: комп'ютерна інженерія, електронний органайзер, тайм-менеджмент, веб-сайт

ABSTRACT

Kushnikov I.O. Software of the personal online organizer system. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi 2021

In this bachelor qualification developed software that is designed for system of personal online organizer has been developed.

The purpose of the work is to create a system of personal online organizer.

The result is a software implementation of a personal online organizer system.

In the process of working on the implementation of the system, research of existing methods, algorithms and software was performed. Developed and implemented its own software, described all its components.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC running Windows 10.

The program is developed in the PHP programming language.

Keywords: computer engineering, electronic organizer, time management, website

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	6
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної бакалаврської роботи.....	6
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	11
2.3 Розгорнута постановка завдання	12
3 ОПИС І ОБґРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	14
3.1 Опис функціонування системи.....	14
3.2 Розробка структурної схеми.....	24
3.3 Розробка функціональної схеми.....	26
3.4 Розробка діаграми процесів	27
4 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	29
4.1 Блок-схеми та опис алгоритмів функціонування системи.....	29
4.2 Захист розробленого програмного забезпечення.....	43
5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	45
6 ОСНОВНІ ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51

КБР-123.21.0054.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Кушніков І.О.			Програмне забезпечення системи особистого онлайн органайзеру	Літ.	Аркуш	Аркушіів
Перев.		Улічев О.С.				Б	1	54
Н.контр.		Гермак В.С.			ЦНТУ КІ-19-2СК			
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- AJAX – Asynchronous Javascript and XML – підхід до побудови інтерактивних користувацьких інтерфейсів веб-додатків.
- CSS – спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.
- CSS3 – Cascading Style Sheets 3 – каскадні таблиці стилів третього покоління.
- HTML – HyperText Markup Language – мова розмітки гіпертексту.
Hypertext Preprocessor – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків.
- ID – Ідентифікатор, унікальна ознака об'єкта або користувача системи.
- JSON – JavaScript Object Notation – текстовий формат обміну даними.
- PDO – PHP Data Objects – розширення для PHP, надає розробнику простий і універсальний інтерфейс для доступу до різних баз даних.
- PHP – скриптова мова програмування, призначена для генерації HTML-сторінок на стороні веб-сервера.
- SQL – Structured query language – мова структурованих запитів.
- БД – База даних.

ВСТУП

Актуальність теми. На сьогоднішній день роль інформаційних технологій у житті людини постійно зростає. Програмне забезпечення застосовується у різних сферах діяльності. Воно допомагає людям у роботі, освіті та відпочинку.

Переходять у цифровий формат різні речі з реального життя, так і паперовий органайзер (записник, блокнот) спочатку отримав нову форму електронного, а потім і онлайн органайзеру. Такі органайзери отримали нові корисні функції порівняно з паперовою версією, зокрема: нагадування про заплановані події, захист особистих даних та синхронізація інформації на різних пристроях. Органайзер є інструментом управління власним часом і допомагає підвищити продуктивність будь-якої діяльності, як особистої, так і професійної.

Багатофункціональні органайзери часто називають також персональними інформаційними менеджерами, що дозволяють декільком користувачам організувати спільне планування і спільну роботу над проектами.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи особистого онлайн органайзеру.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем особистого онлайн органайзеру.
- Дослідження системи особистого онлайн органайзеру.
- Програмна реалізація системи особистого онлайн органайзеру.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі реалізації особистого онлайн органайзеру.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи особистого онлайн органайзеру, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Розроблене програмне забезпечення призначене для накопичення інформації користувача, а потім оперативного пошуку по ній, організації справ і контролю за їх виконанням, відстеження певних користувачем подій. Воно дозволяє вирішувати наступні задачі у цифровому, а не паперовому вигляді:

- планування завдань для контролю за їх виконанням;
- планування подій, прив'язаних до певної дати і часу;
- нагадування про певні події;
- записна книжка і листки-замітки;
- особисті записи (щоденник);
- персональна база знань.

Програми-органайзери призначені для планування побутових і робочих процесів. Вони мають корисні функції: створення заміток, нагадування, контроль часу за вказаний період тощо. Органайзери бувають різного цільового призначення та функціоналу. Розроблене програмне забезпечення є реалізацією універсального персонального органайзера для планування власних справ як особистих, так і ділових.

Управління власним часом, це дія або процес тренування свідомого контролю над часом, витраченим на конкретні види діяльності, при якому спеціально збільшуються ефективність і продуктивність. Управління часом може допомогти отримати ряд навичок, інструментів і методів, використовуваних при виконанні конкретних завдань, проектів і цілей. Цей набір включає в себе широкий спектр діяльності, а саме: планування, розподіл, постановку цілей, делегування, аналіз тимчасових витрат, моніторинг, організацію, складання списків і розстановку пріоритетів. Спочатку управління приписувалося тільки

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

бізнесу або трудовій діяльності, але з часом термін розширився, і почало включати й особисту діяльність. Зазвичай управління часом є необхідністю в розвитку будь-якого проекту (особистого або професійного), оскільки визначає час завершення проекту і масштаб. Органайзери є важливими інструментами для створення системи планування свого часу.

1.2 Область застосування

Ефективна організація часу – актуальна проблема для будь-якої людини, незалежно від її зайнятості та роду діяльності. Таймменеджменту, присвячена не одна книга, регулярно проводяться численні курси, які пропонують різні методики.

Важливим інструментом планування власного часу є електронні та онлайн органайзери. Слід зазначити, що служать вони не тільки для складання розпорядку дня, але і для зберігання даних. Людині складно тримати в голові або ж в різних паперових записниках всю необхідну інформацію, яка повинна бути під рукою.

Це програмне забезпечення допомагає організувати робочий час і відстежувати необхідні користувачу події, а також дозволяє ставити завдання і здійснювати контроль над їх виконанням. Фахівці з тайм-менеджменту часто радять використовувати таке програмне забезпечення людям, що мають проблеми з організацією роботи та дозвілля, тобто тим, хто багато чого не встигає.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи особистого онлайн органайзеру, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної бакалаврської роботи

Перед розробкою програмного онлайн органайзеру важливою задачею постає питання дослідження існуючих аналогів таких систем, а також web-технологій та алгоритмів, які вони використовують. Розглянемо декілька популярних сайтів які можна використовувати як онлайн органайзер.

Trello (<https://trello.com/>) – один із найзручніших сайтів для керування проектами та задачами, який працює на основі японської системи канбан, методу, який спочатку був популяризований Toyota в 1980-х для управління ланцюгами поставок.

У Trello проекти записуються у вигляді дошок, а задачі у вигляді карток на дошках.

Дошка в Trello – це візуальне уявлення проекту або процесу. На такій дошці розміщуються картки, які можуть представляти задачі або об'єкти, впорядковані у списки. Кожен список показує стан будь-якого проекту. Наприклад, при відборі кандидатів на роботу, у першій колонці будуть картки кандидатів, у другій – кандидати, вибрані для співбесіди, в третій – з ким призначена зустріч, у четвертій – з ким зустрілися, а у п'ятій – ті, про кого ви серйозно замислюєтеся, як про свого майбутнього співробітника.

Картки-завдання можна переміщувати як між списками, так і всередині списку, відображаючи поточний стан справ і пріоритетів. Картки мають безліч можливостей. В них можна проводити обговорення, голосування, завантажувати файли даних, задавати “дедлайни”, призначати текстові та колірні мітки.

Основною перевагою є можливість бачити кілька одночасно запущених проектів та їх стан в поточний момент часу. Якщо ви керуєте групою розробників

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

чи інших виконавців, які працюють над проектами з кінцевою датою виконання або фіксованої метою, то дана система може дати вам уявлення про хід проектів у будь-який момент часу. Важливо зауважити, що всі члени робочої групи бачать в реальному часі зміни, що вносяться в проект.

При створенні даного сайту окрім стандартних методів розробки веб-систем було використано функцію бібліотеки jQuery – draggable, що надає можливість перетаскувати DOM елементи будь-куди. Також при користуванні сайтом можна побачити, що після деяких операцій сторінка браузера не перезавантажується, що свідчить про використання AJAX технології, що значно покращує роботу сайту.

Ще однією сильною стороною Trello є наявність повноцінних мобільних клієнтів під iOS і Android, що дозволяють працювати як на смартфонах, так і на планшетах. Недоліком Trello є те що деякі можливості сайту не безкоштовні, а такий інтерфейс сайту представлений англійською мовою, тому що для когось це може стати серйозною перешкодою. При розробці програмного забезпечення системи управління сайтом для керування проектами буде враховано переваги даного сайту та усунено можливість появи його недоліків.

Evernote (<https://evernote.com>) – веб-сервіс і набір програмного забезпечення для створення та зберігання нотаток.

В якості замітки може виступати фрагмент форматowanego тексту, веб-сторінка цілком, фотографія, аудіофайл або рукописний запис. Нотатки можуть також містити вкладення з файлами іншого типу. Нотатки можна сортувати по блокнотам, присвоювати їм мітки, редагувати і експортувати.

Перевагою даного сайту являється те, що Evernote підтримує кілька комп'ютерних і мобільних платформ, включаючи OS X, iOS, Chrome OS, Android, Microsoft Windows, Windows Phone, BlackBerry і webOS, а також пропонує онлайн синхронізацію і резервне копіювання.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Evernote Basic – безкоштовний профіль, що має обмеження за розміром замітки (не більше 25 МБ на кожну замітку) та обсягом трафіку для написання/редагування заміток (60 МБ на місяць).

Evernote Plus – має обмеження за розміром замітки (не більше 50 МБ на кожну замітку) та обсягом трафіку для написання/редагування заміток (1 ГБ на місяць). Включає всі функції Evernote Basic, крім того, підтримує відправку в Evernote до 200 електронних листів на день та має блокування паролем мобільного додатку. Evernote Plus доступний за 300 гривень на рік.

Evernote Premium – не має обмежень на щомісячний обсяг завантажень, але існує обмеження на розмір замітки (200 МБ). Включає в себе всі функції Evernote Plus, а так само забезпечує пріоритетне розпізнавання тексту в зображеннях, додаткові функції безпеки, можливість сканування і оцифрування візитних карток, пошук по тексту усередині файлів Microsoft Office, iWork і документів PDF. Вартість Evernote Premium становить 580 гривень за річну передплату.

Evernote Business – включає в себе всі функції Evernote Premium, крім того, має розширені можливості спільної роботи: єдиний вхід для співробітників, інструменти централізованого адміністрування облікових записів, централізоване зберігання корпоративної інформації. Вартість Evernote Business становить 279 гривень на місяць за одного підключеного співробітника.

Користувачі як безкоштовних, так і платних профілів Evernote можуть надавати доступ до своїх блокнотів іншим користувачам Evernote. Однак платні передплатники можуть відкрити доступ до блокнота не тільки для читання, але й редагування. Для безкоштовних профілів дані на пристроях з iOS і Android в загальному випадку недоступні при відсутності інтернет-з'єднання. Незважаючи на це, деякі замітки кешуються і залишаються доступними, хоча їх редагування може викликати конфлікти при синхронізації. Всі профілі Evernote, як безкоштовні, так і платні, мають максимальне обмеження в 100 000 заміток і 250 особистих блокнотів.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

клієнтів, зберігання бази контактів, обговорень і коментування, реалізована повномасштабна статистика з проектів та виконання завдань.

Перевагою цієї веб-системи є можливість інтеграції з такими сервісами як Office 365 (перегляд і редагування офісних документів в онлайні), Google Drive(прикріплення до завдань документів карток і папок Google Drive), Skype(отримання повідомлень з сайту та відправка відповідей на ці повідомлення), Dropbox(можливість прикріплення документів зі сховища), Google calendar(швидкий експорт завдань з розділу планувальника сайту), Telegram(отримання повідомлень з сайту та відправка відповідей на ці повідомлення). Також є мобільна версія сервісу для комфортної роботи в браузерях.

Сайтом Planfix можна користуватися безкоштовно.

Головною відмінністю сайту, за заявами самої компанії-розробника, є можливість самостійно налаштувати його під роботу організації. Однак у той же час вони досі не прислухалися до прохань клієнтів створити мобільні додатки під iOS і Android, хоча для роботи може вистачити і веб-інтерфейсу.

Mylifeorganized (<http://www.mylifeorganized.net>) – веб-система для колективної роботи. Перевагою є те, що Mylifeorganized підтримує кілька комп'ютерних і мобільних платформ, включаючи Mac OS, BlackBerry, Windows, iPhone, iPad, Adnroid телефон і планшет. Недолік полягає у тому що користування веб-сервісом є платними. На рисунку зображена головна сторінка сайту Mylifeorganized.

Перевагою є простота у користуванні та навчаючі відео, так як це приверне увагу не тільки новачків. Недоліком є низькі можливості в порівнянні з іншими аналогами.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий HTML-код. Це перевага з точки зору безпеки.

У даному проекті PHP буде використовуватись для генерації HTML-сторінок з даними, які буде отримано з бази даних (БД) за допомогою php класу PDO.

5. AJAX – підхід до розробки, який призначений для інтерфейсу користувача, комбінує кілька основних методів і прийомів:

- використання DHTML для динамічної зміни змісту web-сторінки;
- використання XMLHttpRequest для звернення до сервера «на льоту»,

не перезавантажуючи всю сторінку повністю.

Використання цих підходів дозволяє створювати набагато зручніші веб-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія з користувачем.

У даному проекті буде використовуватись для звернення до серверу «на льоту» і отримання інформації з БД з подальшим відтворенням її на сторінку.

6. JSON – це текстовий формат обміну даними. JSON базується на тексті, і може бути з легкістю прочитаним людиною. Формат дозволяє описувати будь-які об'єкти та інші структури даних.

У даному проекті буде використовуватись для опису будь-якої інформації стосовно користувачів, проектів, дошок, а також для реалізації мультимовного сайту.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи особистого онлайн органайзеру.

В процесі розробки кваліфікаційної бакалаврської роботи необхідно виконати наступний обсяг роботи:

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		12

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;
- б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;
- в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;
- г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;
- д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;
- е) провести розрахунки по визначенню економічної ефективності розробленої системи;
- ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;
- з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

введення логіну та E-mail'у, а потім за допомогою технології AJAX створити запит до таблиці з вже зареєстрованими користувачами у БД та перевірити наявність в ній введених даних користувачем. Якщо хоча б один з об'єктів введених користувачем вже зайнятий, то йому можна вивести повідомлення “на льоту”, що введений ним логін або E-mail вже зайнятий. Такі речі перетворюють та налагоджують взаємозв'язок між користувачем та сайтом, що робить користування сайтом більш зручним.

№	Ім'я	Тип
1	id	int(11)
2	UserName	varchar(50)
3	Password	varchar(128)
4	Email	varchar(250)
5	Photo	varchar(150)
6	Active	tinyint(1)

Рисунок 3.1 – Структура таблиці користувачів

Для перевірки достовірності введених даних користувачем було прийнято рішення зробити підтвердження реєстрації через E-mail. Тобто після введення всіх даних і натиснення кнопки реєстрація потрібно надіслати листа на пошту користувача з посиланням на деяку сторінку на сайті, яка буде перевіряти зміст цього посилання і, або активувати профіль користувача, або повідомляти про помилку. Але для цього потрібно створити ще одну таблицю для збереження відповідних даних про користувача та активацію його профіля. Структуру даної таблиці зображено на рисунку 3.2:

- UserID – ідентифікатор користувача для реєстрації;
- SecureKey – код активації профіля;
- Email – E-mail користувача;
- RegConfirm – флаг підтвердження реєстрації;
- ChangePass – флаг зміни паролю профіля.

№	Ім'я	Тип
1	id	int(11)
2	UserID	varchar(128)
3	SecureKey	varchar(128)
4	Email	varchar(250)
5	Change_pass	tinyint(1)
6	RegConfirm	tinyint(1)

Рисунок 3.2 – Структура допоміжної таблиці користувачів

Якщо на сайті існує сторінка реєстрації, то повинна бути і сторінка авторизації користувача, на якій користувач матиме змогу авторизуватись шляхом введення свого логіну та паролю. Далі система повинна перевірити введені дані з таблицею користувачів у БД та, або авторизувати користувача, надавши йому право на перегляд закритої інформації на сайті, або вивести повідомлення про помилку.

Після перегляду аналогів програмного забезпечення системи управління сайтом було вирішено створити мінімальний, простий та зрозумілий інтерфейс користувача.

Так як на сайт будуть заходити користувачі з різних країн світу, було прийнято рішення зробити сайт мультимовним, тобто зробити можливість зміни мови інтерфейсу сайту. Це можна реалізувати у вигляді кнопки з зображенням прапора країни, при натисненні на яку буде змінена мова сайту.

Для реалізації цієї задачі було вирішено створити нову таблицю в базі даних з наступною структурою:

- ID – ідентифікатор слова;
- Word – позначення слова;
- Word_ua – слово на українській мові;
- Word_en – слово на англійській мові;
- Word_ru – слово на російській мові.

Також структуру даної таблиці наглядно зображено на рисунку 3.3.

№	Ім'я	Тип
1	id	int(11)
2	Word	varchar(100)
3	Word_ua	varchar(100)
4	Word_en	varchar(100)
5	Word_ru	varchar(100)

Рисунок 3.3 – Структура таблиці зміни мови інтерфейсу сайту

Дана таблиця міститиме усі слова та фрази, які можна зустріти на сайті на різних мовах. Тоді при зміні на будь-яку мову потрібно завантажити слова з даної таблиці для обраної мови з подальшим виведенням отриманих слів у відповідні блоки сторінки.

Розроблюваний сайт міститиме наступні сторінки:

- головна – головна сторінка сайту, на якій користувач знайомитиметься з інформацією про веб-сайт та його можливостям;
- реєстрація – сторінка, на якій користувач має можливість зареєструватись;
- авторизація – сторінка, на якій користувач має можливість авторизуватись;
- підтвердження реєстрації – сторінка, на яку користувач потрапляє по посиланню з поштового листа під час проходження реєстрації. Дана сторінка перевіряє інформацію з листа та активує профіль користувача;
- зміна паролю користувача – сторінка, на яку користувач потрапляє по посиланню з поштового листа під час відновлення паролю. Ця сторінка надає змогу змінити пароль від профіля користувача на новий;
- сторінка дошок користувача – сторінка, на якій відображаються усі дошки (проекти), які створені користувачем або в яких він приймає участь.
- сторінка дошки – сторінка, на якій розміщується та надаються завдання

по проектам учасникам деякої окремої дошки (проекту);

– профіль користувача – сторінка, на якій буде розміщено інформацію про профіль користувача та його активність у проектах.

На головній сторінці міститься інформація про розроблений сайт з можливістю переходу до сторінки реєстрації та авторизації.

На сторінці з дошками користувач має можливість переглядати створені ним дошки (проекти) або проекти, в яких він приймає участь. На верхній панелі знаходиться кнопка з можливістю відображення бокового меню. На цьому меню знаходяться важливі проекти користувача, останні переглянуті користувачем проекти з можливістю пошуку проекту за назвою. Також на верхній панелі реалізований пошук по користувачам, проектам та завданням. Пошук є динамічним, тобто при кожному натисненні клавіші одразу відображаються результати, які відповідають вказаному критерію. Також на верхній панелі присутня кнопка створення нової дошки та меню профіля користувача з такими функціями:

- перехід до профілю користувача;
- зміна мови інтерфейсу сайту;
- вихід з профіля користувача.

Ще однією функцією верхньої панелі є відображення повідомлень користувача. У даному блоку відображаються усі події пов'язані з користувачем, наприклад, запрошення до участі у новому проекті (дошці).

Нижче верхньої панелі знаходяться усі дошки користувача. У користувача є можливість додати будь-яку з них до важливих дошок, що відразу помічається зірочкою в правому верхньому кутку. При натисненні на цю зірочку користувач автоматично додає проект до списку важливих проектів або видаляє обраний проект зі списку.

Для того щоб зберігати інформацію про дошки потрібно створити дві таблиці в базі даних. Перша таблиця потрібна для збереження інформації про проекти, а друга потрібна для збереження інформації про учасників проектів та їх

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

– Permission – значення, яке відповідає за статус користувача (керівник (1) або працівник (0));

– Last_seen – дата та час останнього перегляду дошки користувачем;

– Starred – значення, яке вказує на приналежність дошки до важливих дошок користувача;

Спочатку відображаються усі важливі дошки користувача, а потім усі інші. Також користувач може створити нову дошку або перейти до сторінки окремої дошки шляхом натиснення на її зображення.

Сторінка окремої дошки містить таку ж саму верхню панель, як і попередня сторінка. Під верхньою панеллю знаходиться інформація про дошку, а саме: назва, позначення важливості дошки зірочкою та тип дошки.

Далі йдуть списки з завданнями (картками). Кожен список має свою назву та деяку кількість завдань. Кожне завдання (картка) представляє собою окремий блок списку, який може містити: назву картки, опис картки, коментарі для даного завдання, прикріплені файли з результатами виконання завдання, зображення аватарів працівників, які працюють над завданням, кольорову стрічку, яка повинна звернути увагу керівника до цього завдання або відобразити вдалий чи невдалий результат виконання завдання.

З перерахованої вище інформації можна зробити висновок, що потрібно створити таблицю для зберігання інформації про списки, які знаходяться в кожному проекті та їх зв'язків, таблицю з інформацією про завдання та зв'язки з їхніми списками, таблицю, яка зберігатиме інформацію про історію активності користувачів на завданнях та дошках в цілому, таблицю інформації про файли, що прикріплюються до кожного завдання, а також таблицю підзавдань до кожного завдання, що будуть відображені у вигляді списку з “чекбоксами”.

Розглянемо структуру кожної вищеописаної таблиці окремо.

На рисунку 3.6 зображено структуру таблиці з інформацією про списки завдань проектів зі зв'язками до кожного проекту.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

№	Ім'я	Тип
1	id	int(11)
2	Name	varchar(100)
3	Board_id	int(11)
4	Position	int(11)
5	Name	int(11)
6	Starred	tinyint(1)

Рисунок 3.6 – Структура таблиці списків з завданнями

- ID – ідентифікатор списку у таблиці;
- Name – назва списку;
- Board_id – ідентифікатор дошки, до якої відноситься список;
- Position – позиція відображено списку на дошці. Так як списки можна міняти місцями за допомогою “drag’n’drop”, потрібно яким-небудь чином зберігати позицію відображення кожного списку у проекті.

Структуру таблиці з інформацією про завдання відображено на рисунку 3.7.

№	Ім'я	Тип
1	id	int(11)
2	List_id	int(11)
3	Name	varchar(100)
4	Description	text
5	Due_date	datetime
6	Members	text
7	Labels	text

Рисунок 3.7 – Структура таблиці завдань зі списку

- ID – ідентифікатор завдання у таблиці;
- List_id – ідентифікатор списку, до якого відноситься завдання;
- Name – назва завдання;
- Description – опис завдання;

- Due_date – кінцева дата виконання завдання;
- Members – рядок, в якому будуть перераховані ідентифікатори користувачів, які працюють над завданням;
- Labels – рядок, що містить кольори стрічок, які прикріпленні до даного завдання.

Дана таблиця являється найважливішою, тому що містить дуже багато основної інформації, а також зв'язків з іншими таблицями.

На рисунку 3.8 зображено структуру таблиці з інформацією про історію активності користувачів у завданні або на дошці в цілому.

№	Ім'я	Тип
1	id	int(11)
2	User_id	int(11)
3	Text	text
4	Type	varchar(10)
5	Time	datetime

Рисунок 3.8 – Структура таблиці історії активності користувачів

- ID – ідентифікатор події у таблиці;
- User_id – ідентифікатор користувача події;
- Text – опис події;
- Type – тип події;
- Time – дата та час створення події користувачем.

Подією може бути залишений користувачем коментар до завдання, прикріплення файлу до завдання, переміщення завдання до іншого списку, прикріплення стрічки до завдання, вказання працівника, який працюватиме над завданням, виконання якого-небудь підзавдання зі списку підзавдань.

Структуру таблиці інформації про файли, що користувач може прикріпити до завдання зображено на рисунку 3.9.

№	Ім'я	Тип
1	id	int(11)
2	Task_id	int(11)
3	Name	varchar(100)
4	Date	datetime

Рисунок 3.9 – Структура таблиці прикріплених файлів до завдання

- ID – ідентифікатор файлу у таблиці;
- Task_id – ідентифікатор завдання до якого прикріплено файл;
- Name – назва файлу, також назва файлу служить шляхом до файлу для завантаження;
- Date – дата та час прикріплення файлу до завдання.

Структуру таблиці підзавдань завдання зображено на рисунку 3.10.

№	Ім'я	Тип
1	id	int(11)
2	Task_id	int(11)
3	Description	text
4	Completed	tinyint(1)

Рисунок 3.10 – Структура таблиці підзавдань

- ID – ідентифікатор підзавдання у таблиці;
- Task_id – ідентифікатор завдання у якому існує підзавдання;
- Description – опис підзавдання;
- Completed – флаг, що вказує на виконане чи не виконане завдання.

Підсумувавши все вищенаведене можна зробити висновок, що на сторінці завдання користувач має можливість вказати опис завдання, додати коментар до картки з прикріпленням файлів, вказати декількох працівників на виконання вибраного завдання, додати кольорову стрічку до завдання, таких чином звернувши увагу до цього завдання або помітивши, що дане завдання виконано

гарно чи погано. Також користувач може створити список з підзавдань для реалізації поставленого завдання і встановити кінцеву дату на його виконання. Існує можливість перемістити картку з одного списку в інший або перемістити увесь список з картками. Також є можливість позначити важливість та тип завдання.

У правій частині екрану розташоване бокове меню дошки, яке відображає учасників дошки або проекту з можливістю додавання нових учасників до цієї дошки, якщо користувач має привілегію адміністратора даної дошки.

Також це меню має ряд наступних функцій:

- зміна кольору фону дошки;
- можливість змінити права учасника проекту;
- відображення активності діяльності учасників проекту.

3.2 Розробка структурної схеми

Структурна схема сайту – це сукупність об'єктів та частин сайту та взаємозв'язки між ними. Призначенням структурної схеми є наглядне відображення складових частин сайту, його основних блоків, вузлів та взаємозв'язок між ними.

Структурну схему розроблюваного сайту зображено на рисунку 3.11.

Зі схеми можна побачити, що сайт має головну сторінку з якої користувач може перейти до сторінки реєстрації, авторизації та до сторінки адміністративної панелі сайту. Зі сторінки реєстрації користувач може потрапити на сторінку авторизації. На сторінці авторизації користувач має можливість перейти на сторінку відновлення паролю, якщо він забув свій пароль для входу. Також можна потрапити до сторінки усіх записів користувача. Звідси користувач може перейти до сторінки окремого запису або до сторінки свого профілю. Також на цій сторінці користувач може вийти зі свого профілю.

Зі сторінки окремого запису користувач може перейти до відображення

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

список дошок за темою запису, що містять його справи та плани. Кожна дошка також містить всередині картки з конкретними задачами та можливістю встановлення нагадування по ним.

Користувач може створювати, редагувати, переглядати та видаляти дошки та картки, а також групувати їх у списки, що відображають різні напрямки його діяльності.

Також користувач має можливість редагування даних власного профілю та здійснення зміни налаштувань органайзеру.

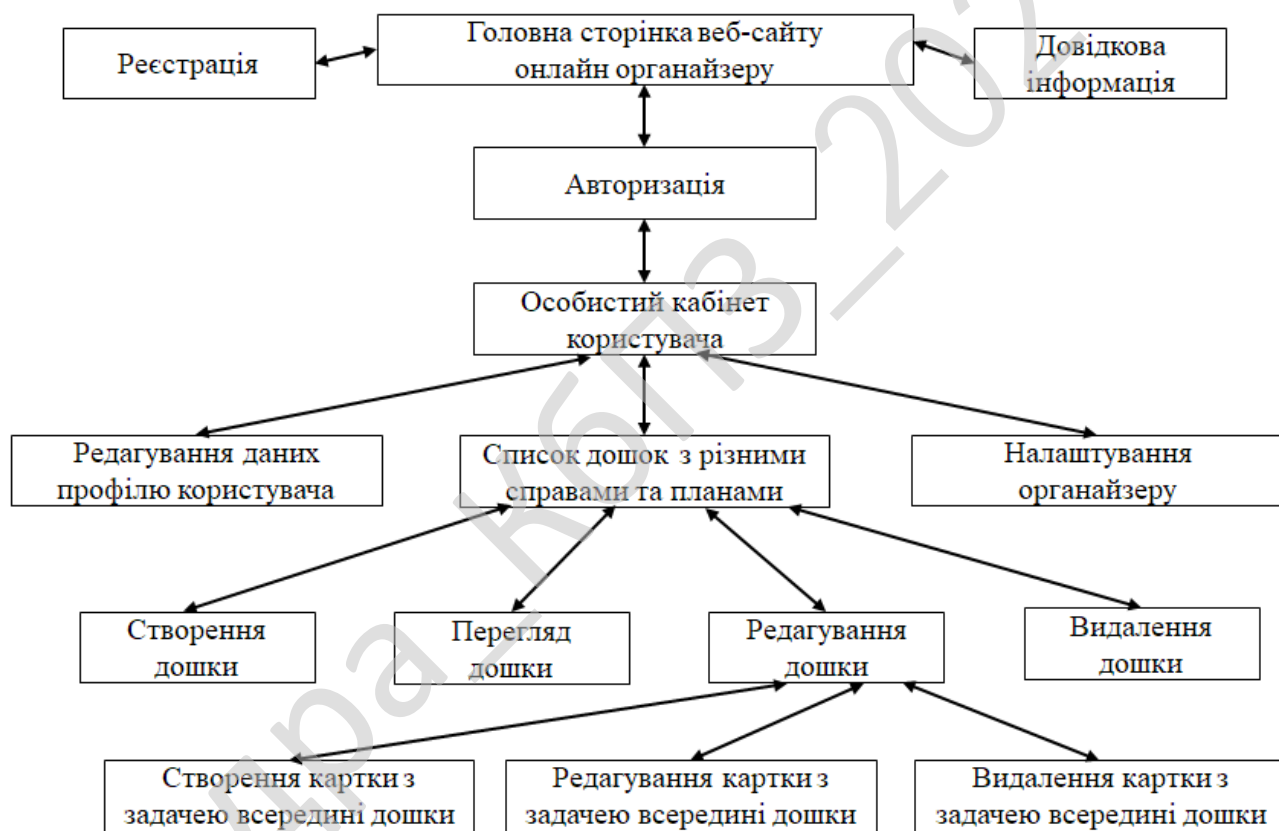


Рисунок 3.11 – Структурна схема сайту

3.3 Розробка функціональної схеми

Функціональну схему сайту зображено на рисунку 3.12. Функціональна схема складається з таких блоків:

- модуль головної сторінки;
- модуль бази даних;
- модуль кабінету користувача.

Розглянемо кожен з блоків окремо.

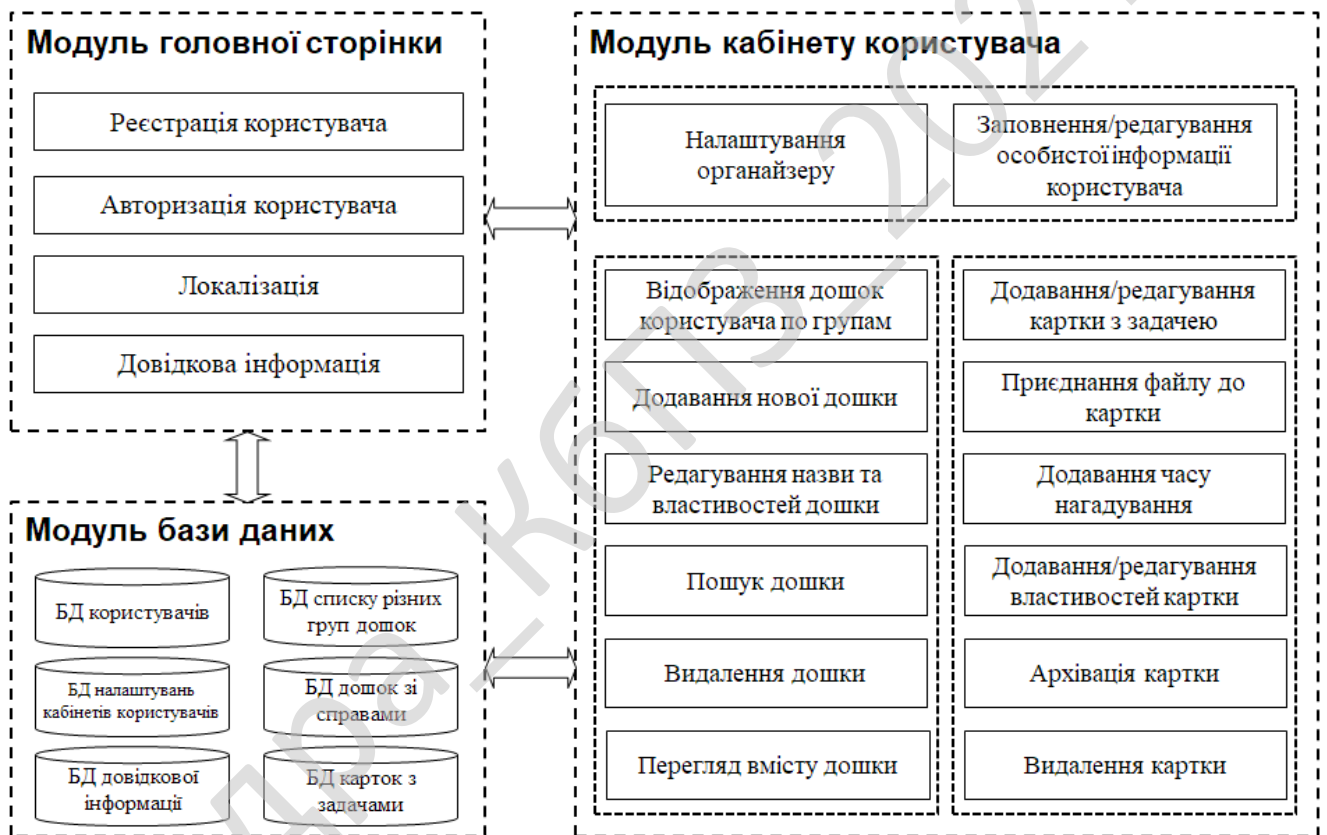


Рисунок 3.12 – Функціональна схема сайту

Модуль головної сторінки реалізує функції реєстрації та авторизації користувача, а також забезпечує локалізацію веб-сайту та виведення довідкової інформації про систему.

Модуль бази даних містить бази даних системи та забезпечує роботу з

ними. Система має наступні бази даних:

- База даних користувачів системи.
- База даних налаштувань кабінетів користувачів.
- База даних довідкової інформації.
- База даних списку різних груп дошок.
- База даних дошок зі справами.
- База даних карток з задачами.

Модуль кабінету користувача дозволяє змінювати налаштування органайзеру, переглядати та редагувати особисті дані, а також реалізує функції системи контролю часу користувача, що надають наступні можливості:

- Відображення дошок користувача по групам.
- Додавання нової дошки.
- Редагування назви та властивостей дошки.
- Пошук дошки.
- Видалення дошки.
- Перегляд вмісту дошки.
- Додавання/редагування картки з задачею.
- Приєднання файлу до картки.
- Додавання часу нагадування.
- Додавання/редагування властивостей картки.
- Архівація картки.
- Видалення картки.

3.4 Розробка діаграми процесів

Діаграму процесів зображено на рисунку 3.13.

Після переходу на головну сторінку користувач може пройти авторизацію або зареєструватись, якщо він ще не має свого профіля. Він також може відновити свій пароль перейшовши на сторінку відновлення паролю з форми

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

авторизації.

Будучи авторизованим користувач має змогу перейти до сторінки своїх дошок справ. На цій сторінці користувач може переглянути дошки, виконати пошук по дошках за ім'ям створити нову дошку або видалити певну дошку.

На сторінці окремої дошки користувач може спостерігати список карток з задачами по даній дошці. Він також має можливість створювати нові картки, міняти місцями картки, редагувати, архівувати та видаляти їх..

На сторінці завдання користувач має можливість вказати опис завдання, додати коментар до завдання з прикріпленням файлів, додати кольорову стрічку до завдання, відмітити факт виконання завдання.

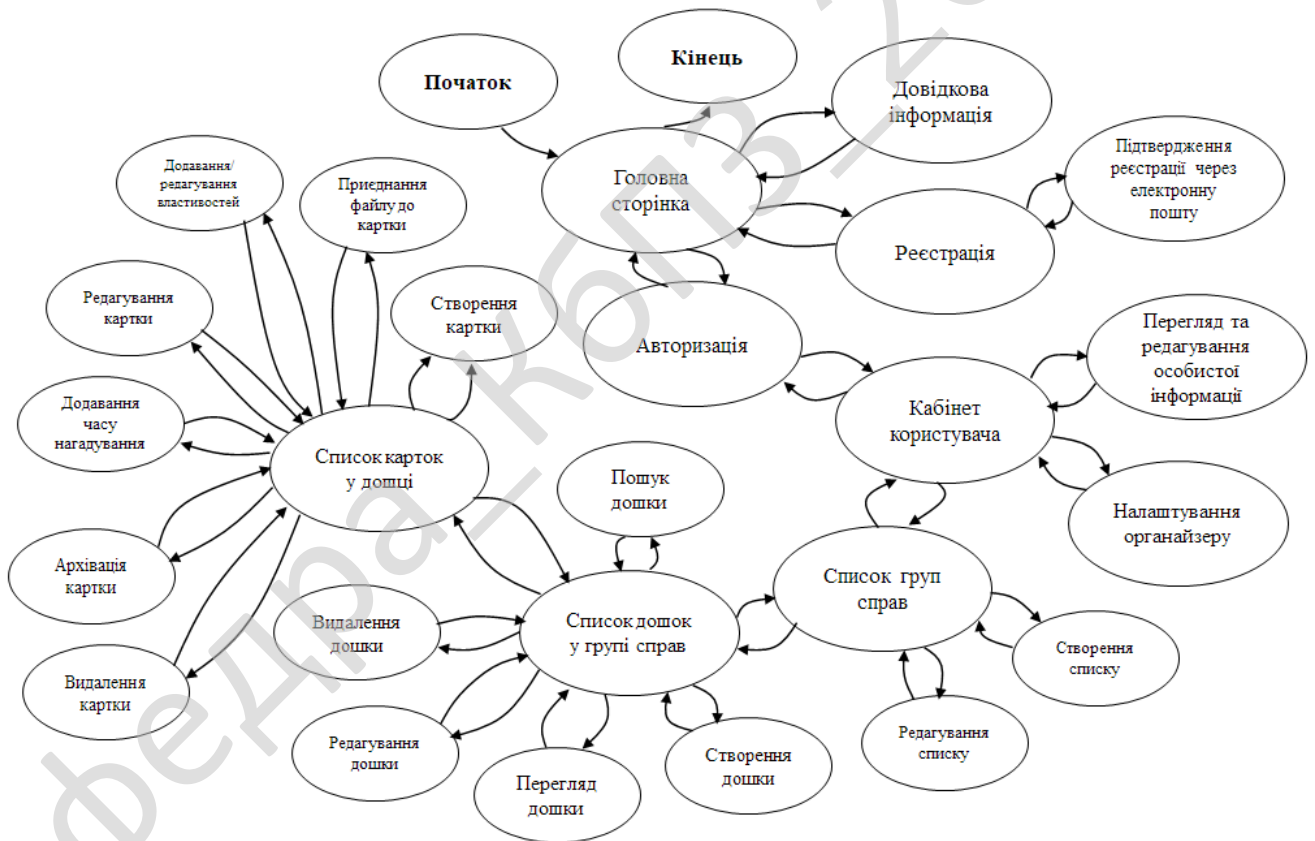


Рисунок 3.13 – Діаграма процесів

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Як тільки користувач потрапив на головну сторінку сайту він повинен пройти авторизацію або реєстрацію у випадку існування його профіля. На рисунку 4.1 зображено блок-схему алгоритму роботи основної програми.

На сторінці реєстрації користувач має можливість пройти реєстрацію нового профіля для доступу до закритої частини сайту. Для успішної реєстрації користувач повинен надати свою фотографію, ім'я, поштову адресу та бажаний пароль. Після введення всіх даних система перевіряє заповненість усіх полів та присутність завантаженого зображення і, або виводить помилку, якщо щось було пропущено, або створює два записи у таблицях бази даних. Перший запис містить введену користувачем інформацію про його профіль, а другий містить інформацію потрібну для активації даного профіля, що зберігається у таблиці з інформацією про профілі, що очікують активації. Далі система надсилає листа на введену користувачем поштову адресу з посиланням, яке містить код активації профіля і повідомляє користувача про те, що на вказану ним адресу було відправлено листа з кодом активації.

Після того як користувач перейшов по отриманому посиланню система перевіряє інформацію у цьому посиланні, а саме код активації профіля і якщо даний код буде збігатися з кодом у рядку в таблиці з неактивованими профілями – система видалить інформацію про неактивований профіль користувача та встановить, що даний профіль користувача активовано.

Потім система перенаправляє користувача на сторінку авторизації, на якій він має можливість увійти на свій профіль та розпочати роботу.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

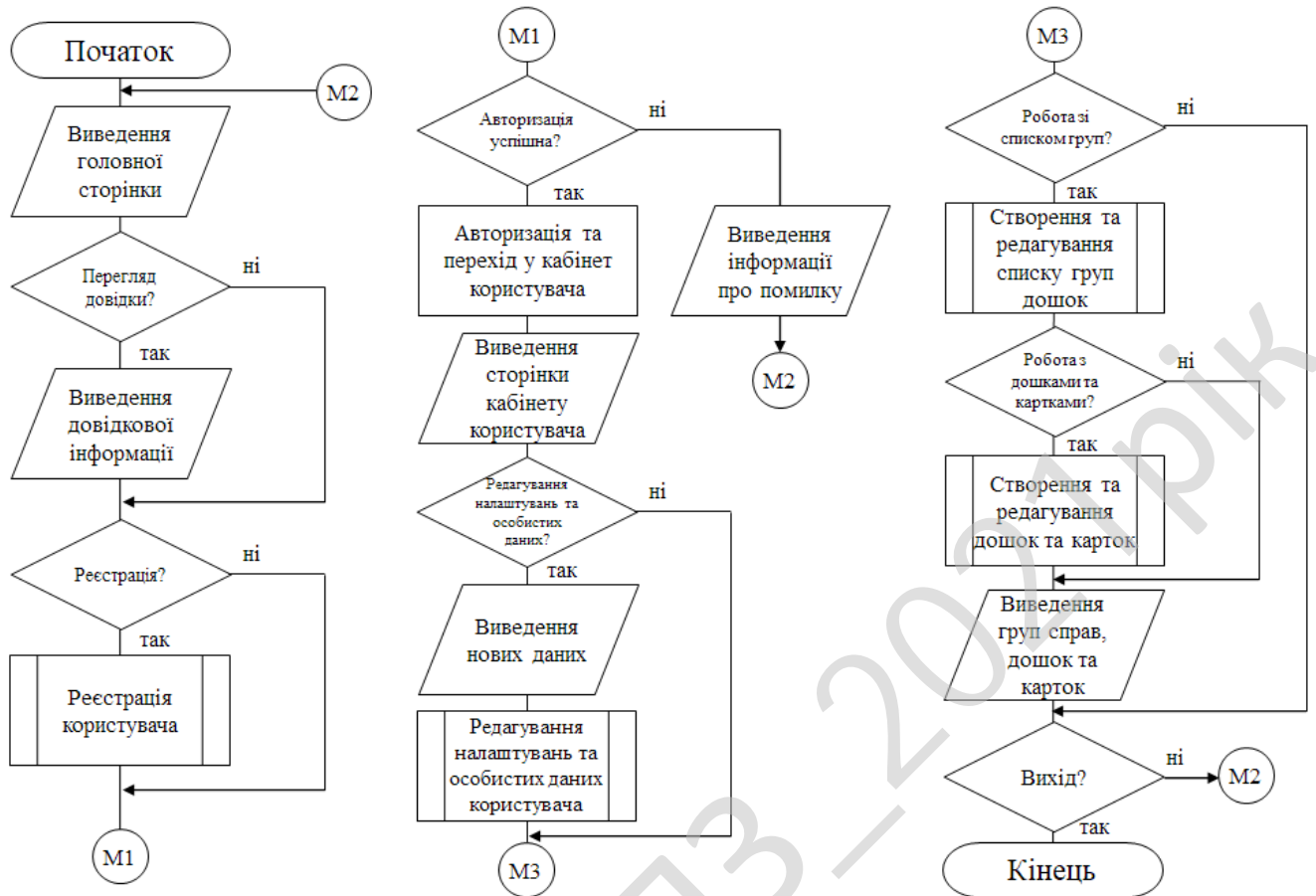


Рисунок 4.1 – Блок-схема роботи основної програми

Наступний блок коду відповідає за реалізацію вищенаведеного алгоритму реєстрації користувача у системі:

```
// Обробник кнопки завантаження зображення
$('.btn-file :file').on('fileselect', function(event, numFiles, label) {
    var _validFileExtensions = [".jpg", ".jpeg", ".png"];
    var sFileName = $("#photo_field").val();
    // Перевірка розширення зображення
    if (sFileName.length > 0) {
        var blnValid = false;
        for (var j = 0; j < _validFileExtensions.length; j++) {
            var sCurExtension = _validFileExtensions[j];
            if (sFileName.substr(sFileName.length - sCurExtension.length,
                sCurExtension.length).toLowerCase() == sCurExtension.toLowerCase()) {
                blnValid = true;
                break;
            }
        }
    }
});
```



```

        return false;
    }
    // Перевірка на зайнятий логін
    check('login');
    if(event.keyCode == '13'){ $("#register").click(); }
});
// Обробник натиснення клавіші у полі e-mail'у
$("#email_field").keyup(function(event) {
    var email = $("#email_field").val();
    // Перевірка на валідний e-mail
    if(!/^((^[^<>() []\.\.,;:\s@\"']+(\.[^<>() []\.\.\.,;:\s@\""]+)*|(\\".+\"))@(\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\)|([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/ .test(email)){
        $(".email_message").html("Не дійсний E-mail!");
        can_register.email = false;
        return false;
    }
    // Перевірка на зайнятий e-mail
    check('email');
    if(event.keyCode == '13'){ $("#register").click(); }
});
// Обробник натиснення клавіші у полі паролю
$("#password_field").keyup(function(event) {
    var pass = $("#password_field").val();

    if(pass.length < 6){
        $(".password_message").html("Занадто короткий пароль!");
        can_register.pass = false;
    } else {
        $(".password_message").html('');
        can_register.pass = true;
    }
    if(event.keyCode == '13'){ $("#register").click(); }
});
// Функція перевірки на зайняті значення у таблиці БД
function check(field) {
    var value = $("#" + field + "_field").val(),
        errors = {login: "Логін вже зайнятий!", email: "E-mail вже зайнятий!"};
    $.ajax({
        type: "POST",
        url: "ajax/register.php",

```

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

КБР-123.21.0054.00.00.ПЗ

Арк.

32

```

        data: {
            check : true,
            field : field,
            value : value
        }
    }).done(function(msg) {
        if(msg == 'false'){
            $(". " + field + "_message").html(errors[field]);
            can_register[field] = false;
        } else {
            $(". " + field + "_message").html('');
            can_register[field] = true;
        }
    });
}

// Функція валідації всіх полів при відправці форми
function validate() {
    var login = $("#login_field").val(),
        email = $("#email_field").val(),
        pass = $("#password_field").val(),
        photo = $("#photo_field").val();
    if(login == '' || email == '' || pass == '' || photo == '') {
        showError("Заповніть порожні поля!", false);
        return false;
    }
    if(!(can_register.login && can_register.email && can_register.pass &&
can_register.photo)) {
        showError("Перевірте заповнені поля на правильність!", false);
        return false;
    }
    $(".error_notification").css({"height": "0px"});
    return true;
}

// Файл для AJAX-запитів для перевірки на зайнятий логін або e-mail
$config = include "../config/db.php";
if(isset($_POST['check'])){
    $field = $_POST['field'];
    $value = $_POST['value'];
    $col = '';
    if($field == 'login'){ $col = 'username'; } else { $col = 'email'; }
    try{
        $dbh = new PDO("mysql:host = $config[host]; dbname =

```

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

КБР-123.21.0054.00.00.ПЗ

Арк.

33


```

        try {
            $dbh = new PDO("mysql: host = $config[host]; dbname =
$config[db_name]", $config['username'], $config['password']);
            $sth = $dbh->prepare("INSERT INTO users (username,
email, password, active, photo) values ('$login', '$email', '$password', 0,
'$file_name')");

            $sth->execute();
            $userid = $dbh->lastInsertId();
            $key = $login . $email . date('Y-m-d H:i:s');
            $key = md5($key);
            $sth = $dbh->prepare("INSERT INTO confirm VALUES (NULL,
'$userid', '$key', '$email', 0, 1)");
            $sth->execute();
            require("assets/class.phpmailer.php");
            $mail = new PHPMailer();
            $mail->CharSet = 'UTF-8';
            $mail->From      = "noreply@pandya.esy.es";
            $mail->FromName = "Pandya";
            $mail->AddAddress($email);
            $mail->Subject  = $words->email_subject;
            $mail->Body      = $words->email_body .
"\n\nhttp://pandya.esy.es/confirm.php?username=" . $login . "&email=" . $email .
"&secure_key=" . $key;

            echo "<script> showError(\"Протягом хвилини на " .
$email . " буде надіслано листа з кодом підтвердження реєстрації!\", true);
            $(\"#login_field\").val('');
            $(\"#email_field\").val('');
            $(\"#password_field\").val('');
            </script>";
        }
    } catch (PDOException $e){
        echo $e->getMessage();
    }
    $dbh = null;
}
} else {
    echo "<script> showError(\"Недопустимий файл.\", false);</script>";
}
}
}

```

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

КБР-123.21.0054.00.00.ПЗ

Арк.

35

Після вводу даних у відповідні поля система перевіряє записи у таблиці бази даних на наявність збігу логіну та паролю з введеними даними від користувача. Якщо співпадінь не було виявлено користувачу виводиться помилка, інакше – система перевіряє профіль користувача на активацію. Якщо профіль не було активовано, то користувачу виводиться помилка, а в іншому випадку система зберігає інформацію про користувача в сесії та створює відповідний запис у “куках” браузера, що існують не більше одного дня. Це потрібно для того, щоб користувач не проходив авторизацію кожного разу, а лише один раз на день, що дуже зручно. Користувач матиме можливість вийти зі свого профіля у будь-який час. Також на сторінці авторизації є можливість відновити свій пароль, якщо він був загублений. У такому разі система надсилає листа на пошту користувача з посиланням на сторінку на веб-сайті, перейшовши на яку він матиме можливість змінити свій старий пароль на новий. Це було зроблено для того, щоб пересвідчитись, що зміну паролю вимагає справжній власник профіля, а не злоумисник.

Наступний блок коду відповідає за реалізацію алгоритму авторизації користувача:

```
// Обробник кнопки авторизації
$("#login_signup").click(function() {
    // Отримуємо значення полів логіну та паролю
    var username = $("#login_username").val(),
        password = $("#login_password").val();
    // Якщо вони пусті виводимо помилку
    if(username == '' || password == '') {
        $('.login_error_confirm').css({'height': '35px', 'background-color':
'#F9D8D3', 'border-left': '6px solid #F9756B'});
        $('.login_confirm_message').html("Заповніть порожні поля!");
        return false;
    } else {
        $('.login_error_confirm').css({'height': '0px', 'background-color':
'#fff', 'border-left': '0px', 'margin': '0 auto'});
        $('.login_confirm_message').html('');
    }
    // Інакше виконуємо AJAX-запит до файлу check_user.php з даними
    $.ajax({
```

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

```

type: "POST",
url: "ajax/check_user.php",
data: {
    check_user: true,
    username: username,
    password: password
}
// Функція виконується при завершенні AJAX-запиту
}).success(function(msg) {
    // Якщо прийшла відповідь, що такого користувача не існує, виводимо
помилку
    if(msg == 'false') {
        $('<div class="login_error_confirm">').css({'height': '50px', 'background-
color': '#F9D8D3', 'border-left': '6px solid #F9756B'});
        $('<div class="login_confirm_message">').html("Неправильний логін або
пароль!");
    } else {
        // Інакше переходимо до сторінки з проектами
        window.location.href = "/me";
    }
});
});
// Обробник AJAX-запитів check_user.php
session_start();
$config = include "../config/db.php";
if(isset($_POST['check_user'])) {
    $user = $_POST['username'];
    $pass = $_POST['password'];
    try {
        $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);
        $sth = $dbh->prepare("SELECT * FROM users WHERE username = '$user'");
        $sth->execute();
        $sth->setFetchMode(PDO::FETCH_ASSOC);
        $row = $sth->fetch();
        if($row['id'] != '') {
            if (password_verify($pass, $row['password']) && $row['active'] == 1)
{
                echo 'true';
                $_SESSION['is_logged_in'] = true;
                $_SESSION['login'] = $user;
                setcookie("login", $user, strtotime( '+30 days' ), "/", "", "",

```

						КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			37

```

TRUE);
        setcookie("id", $row['id'], strtotime( '+30 days' ), "/", "",
        "", TRUE);
    }
}
echo 'false';
} catch (PDOException $e){
    echo $e->getMessage();
}
$dbh = null;
}

```

Після реєстрації та успішної авторизації у системі користувач одержує доступ до свого кабінету користувача та можливість здійснювати записи і планувати свою діяльність в електронному органайзері. Система планування часу реалізована за допомогою дошок зі справами та карток з задачами і нагадуваннями.

На рис. 4.2 наведено блок-схему алгоритму роботи з дошками справ та планів.

Коли користувач переходить до вікна роботи з дошками, він може створювати, редагувати та видаляти дошки. Також він може здійснювати пошук по дошкам за пошуковими фразами.

Під час редагування дошки до неї можна додавати картки з задачами. Картки можуть мати кольорову стрічку для візуального розділення їх за типами, а також нагадування, що спрацює як будильник в потрібний час, щоб нагадати про необхідність виконання певної справи. Картки також можна редагувати, видаляти, а також архівувати і змінювати місцями.

На сторінці дошок користувача зображено дошки (проекти), які були створені користувачем.

Спочатку система перевіряє наявність авторизованого користувача у сесії або у “cookie” файлах. Якщо користувач не авторизований на сайті – система перенаправляє його на сторінку авторизації, інакше – система надсилає запит до таблиці “users”, для того щоб дістати інформацію про користувача: його ім’я,

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

аватар і інше. Далі система виводить отримані дані у верхній частині екрану. Потім система надсилає запит до таблиці “user_boards” бази даних, щоб дістати інформацію про дошки користувача. Потім система виводить усі дошки до відповідного блоку на сторінці веб-сайту і починає очікувати дію користувача.

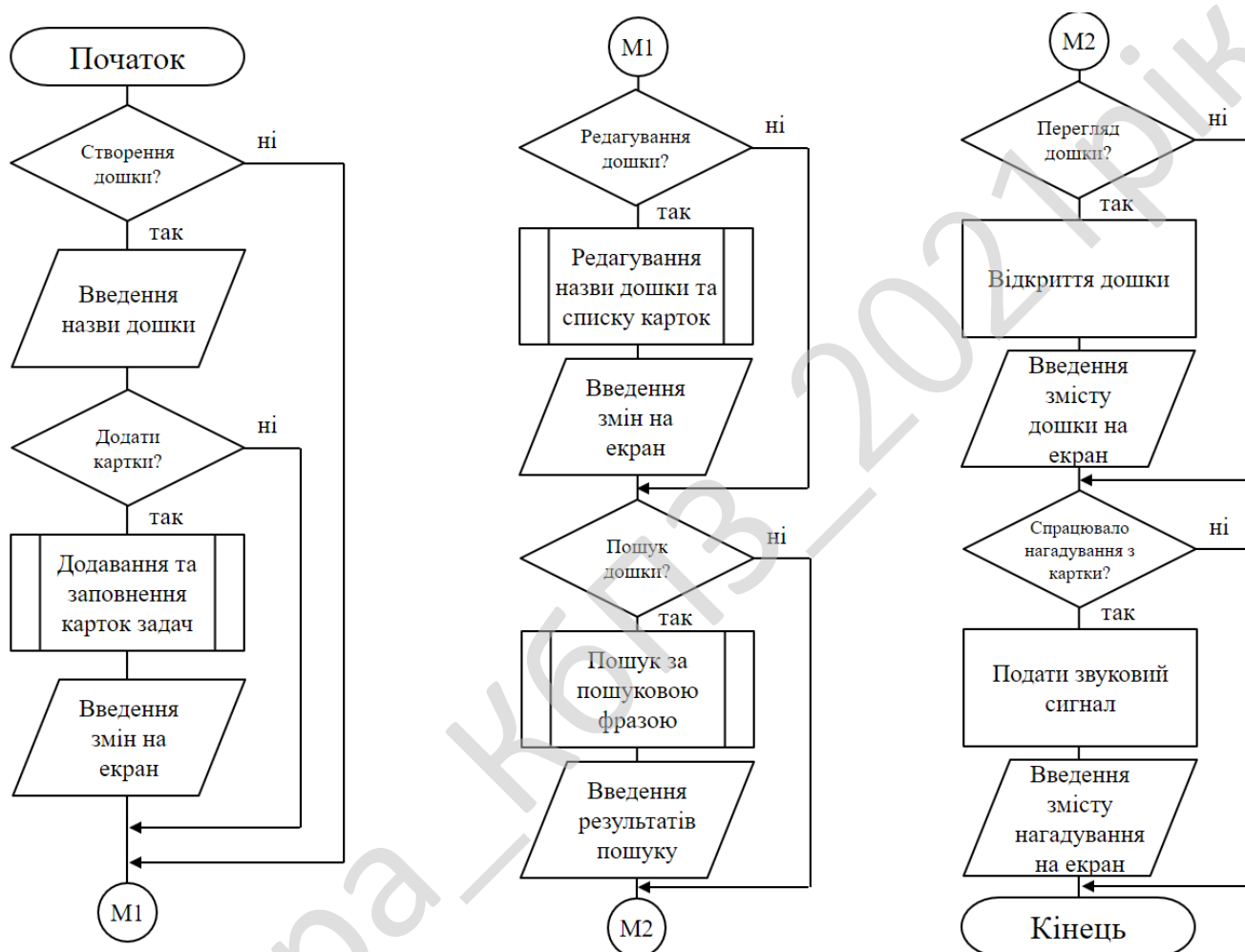


Рисунок 4.2 – Блок-схема алгоритму роботи з дошками справ та планів

Наступний лістинг відображає реалізацію вищенаведених дій:

```

<?php
session_start();
// Перевірка на авторизованого користувача
if($_SESSION['is_logged']){
    $login = $_SESSION['login'];
} else if($_COOKIE['login'] != ''){
    $login = $_COOKIE['login'];
}
  
```

```

}
// Перенаправлення до сторінки авторизації
if(!$login) {
    header( 'Location: /login', true, 307);
}
// Отримання ідентифікатора користувача
$user_id = $_COOKIE['id'];
$config = include "config/db.php";
$dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);
// Отримуємо інформацію про користувача
$stmt = $dbh->prepare("SELECT * FROM users WHERE id=:user_id");
$stmt->execute(array('user_id' => $user_id));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
$photo = $row['photo'];
$username = $row['username'];
?>
<!DOCTYPE html>
<html lang="en">
<body>
    <div class="container-fluid">
        <?php include 'blocks/header_line.php' ?>
        <div class="main_container row">
            <div class="boards_container">
                <?php // Зчитуємо інформацію про проекти користувача
                $stmt = $dbh->prepare("SELECT board_id, last_seen, starred FROM
user_boards WHERE user_id=:user_id");
                $stmt->execute(array('user_id' => $user_id));
                $boards = array();
                while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                    array_push($boards, array("id" => $row['board_id'],
"last_seen" => $row['last_seen'], "starred" => $row['starred']));
                }
                $user_board_ids = array();
                $starred_boards_ids = array();
                // Отримуємо інформацію про важливі дошки користувача
                foreach ($boards as $board) {
                    array_push($user_board_ids, $board["id"]);
                    $starred_boards_ids[$board["id"]] = $board["starred"];
                }
                $user_board_ids = join(',', $user_board_ids);
                $stmt = $dbh->query("SELECT * FROM boards WHERE id IN

```

						КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			40

```

($user_board_ids) ORDER BY id DESC");
    $all_boards = array();
    $starred_boards = array();
    while($row = $sth->fetch(PDO::FETCH_ASSOC)) {
        if($starred_boards_ids[$row['id']] == "1") {
            array_push($starred_boards, $row);
        }
        array_push($all_boards, $row);
    }
    // Виведення важливих дошок на екран користувача
    $starred_boards_display_value = count($starred_boards) > 0 ?
"inline-block" : "none";
    echo '<div class="starred_boards" style="display: ' .
$starred_boards_display_value . '">
        <div class="boards_heading">
            <i class="boards_heading_icon fa fa-star-o"> <span
class="boards_heading_text">Важливі дошки</span></i>
        </div>';
    if(count($starred_boards) > 0) {
        foreach ($starred_boards as $board) {
            echo '<div class="col-sm-3">
                <div class="board" id="' . $board['id'] . '">
                    <a class="open_board" href="/board?show=' .
$board['id'] . '" style="background-color: ' . $board['background'] . ';">
                        <span class="board_details">
                            <span class="board_title">' . $board['name'] .
'</span>
                        </span>
                    <i class="star_icon board_starred fa fa-star"></i>
                    </a>
                </div>
            </div>';
        }
    }
    echo '</div>';
    // Виведення усіх проектів користувача
    echo '<div class="all_boards">
        <div class="boards_heading">
            <i class="boards_heading_icon fa fa-clone"> <span
class="boards_heading_text">Мої дошки</span></i>
        </div>';
        foreach ($all_boards as $board) {

```

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На головній сторінці розробленого веб-сайту онлайн організатору відображено коротку інформацію про розроблений сайт. Для того щоб розпочати управління проектами потрібно бути авторизованим користувачем. Пройти авторизацію можна натиснувши на кнопку “Увійти”. Після натиснення на цю кнопку буде відображено сторінку авторизації, яку зображено на рисунку 5.1.

Авторизація

Ваше Ім'я

Пароль

Увійти

[Забули пароль?](#)

[Реєстрація](#)

Рисунок 5.1 – Фрагмент сторінки авторизації користувача

Після введення логіну та паролю система перевірить введену користувачем інформацію, проведе авторизацію та дозволить переглядати та створювати дошки. Якщо у користувача ще не має зареєстрованого профіля, то він може зареєструвати його натиснувши на кнопку “Реєстрація”. Для реєстрації нового користувача потрібно завантажити свою фотографію, ввести своє ім'я, E-mail та пароль. Сторінку реєстрації зображено на рисунку 5.2.

Реєстрація

Ваше фото

Ваше ім'я

Ваш E-mail

Пароль

Рисунок 5.2 – Фрагмент сторінки реєстрації нового користувача

Якщо в яких-небудь полях буде введено інформацію неналежним чином, система виведе повідомлення про помилку у відповідному полі.

Тобто після введення всіх даних система надсилає листа на введену пошту для підтвердження реєстрації. У отриманому листі знаходиться посилання, перехід по якому повинен активувати профіль. Як тільки користувач переходить за посиланням, система перевіряє інформацію у посиланні та активує профіль користувача.

Після того як профіль активовано, користувач може авторизуватись на сайті і перейти до управління проектами. На сторінці проектів користувача, що зображено на рисунку 5.3, містяться проекти, які були створені користувачем або проекти, в яких він приймає участь.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

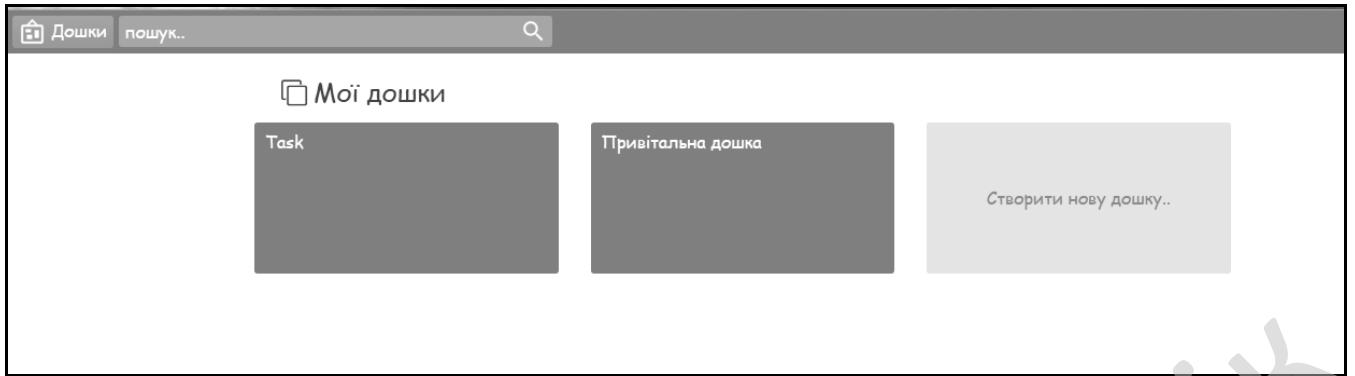


Рисунок 5.3 – Фрагмент сторінки проектів користувача

Для того щоб створити нову дошку потрібно натиснути на кнопку “Створити нову дошку”. Після натиснення на цю кнопку потрібно ввести назву дошки у форму, яка зображена на рисунку 5.4.

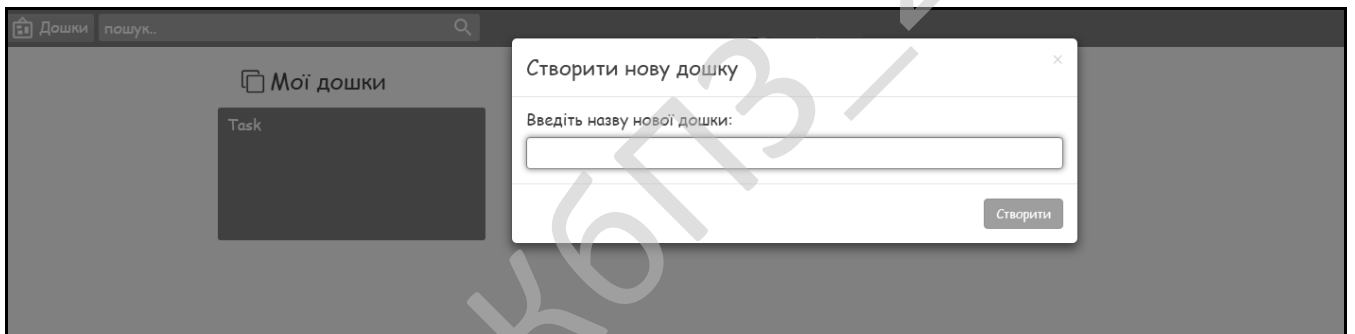


Рисунок 5.4 – Форма заповнення для створення нової дошки

На сторінці вибраного проекту можна створювати списки и додавати до них завдання. Для того щоб змінити тип дошки (приватна або публічна) потрібно відкрити дошку та натиснути кнопку “Публічна”, таким чином публічна дошка стає приватною. Це можна побачити на рисунку 5.5.



Рисунок 5.5 – Зміна приватності дошки

На верхній панелі можна відкрити бокове меню, в якому відображаються “Важливі дошки”, “Мої дошки” та є можливість пошуку по назві дошки.

Для того щоб позначити важливість дошки потрібно натиснути на позначення “зірочки”. Це можна побачити на рисунку 5.6.

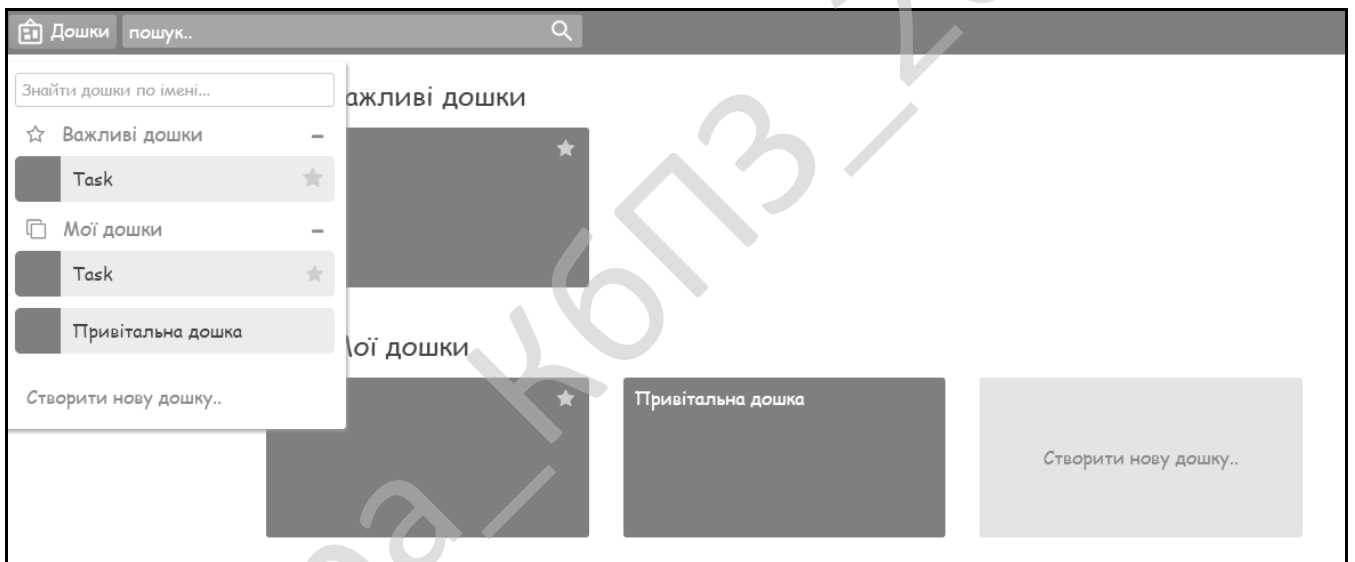


Рисунок 5.6 – Фрагмент сторінки з проектами користувача

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної бакалаврської роботи, призначено для системи особистого онлайн органайзеру.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем особистого онлайн органайзеру.
- Досліджена система особистого онлайн органайзеру.
- На основі отриманих результатів досліджень створена програмна реалізація системи особистого онлайн органайзеру.

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання особистого онлайн органайзеру.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня РНР. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи особистого онлайн органайзеру. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ВСрут.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Кафедра КБПЗ – 2021 рік

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. М.: ДМК Пресс, 2002.
2. Приемы объектно-ориентированного проектирования, из-во “ДМК”, 376 стр., 2011р.
3. Регулярные выражения. 3-е издание, из-во “Символ-Плюс”, 608 стр., 2008р.
4. Разгони свой сайт. Методы клиентской оптимизации веб-страниц. Николай Мацеевский, 264 стр., 2009р.
5. JavaScript. Подробное руководство, 5-е издание, из-во “Символ-Плюс”, 992 стр., 2009р.
6. jQuery. Подробное руководство по продвинутому JavaScript, 2-е издание, из-во “Символ-Плюс”, 624 стр., 2011р.
7. PHP 5, из-во “ВНУ-СПб”, 1104 стр., 2008р.
8. CSS – каскадные таблицы стилей. Подробное руководство, 2-е издание, из-во “Символ-Плюс”, 576 стр., 2005р.
9. “Алан Купер об интерфейсе. Основы проектирования взаимодействия”, из-во “Символ-Плюс”, 688 стр.
10. Паронджанов В. Учись писать, читать и понимать алгоритмы / В. Паронджанов. - М.: ДМК Пресс, 2016. - 458 с.
11. Уайт Э. PHP 5 на практике - НТ Пресс, 2016. - 271 с.
12. Котеров Д., Костарев А. PHP 5 В подлиннике. – СПб: БХВ-Петербург, 2016. – 1104.
13. Вигерс К. Разработка требований к программному обеспечению / Карл Вигерс , Джой Битти. - М.: Русская Редакция, БХВ-Петербург, 2016. - 736 с.
14. Ахо А.В. Структуры данных и алгоритмы / Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман. - М.: Вильямс, 2016. - 400 с.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

15. Трофимов В. В. Алгоритмизация и программирование : учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2019. — 137 с.

16. Берд Дж. Веб-дизайн.Руководство разработчика. / Дж. Берд. - СПб.: Питер, 2012. - 224 с.

17. Гарретт, Д. Веб-дизайн. Элементы опыта взаимодействия / Д. Гарретт. - СПб.: Символ-плюс, 2015. - 192 с.

18. Голомбински К. Добавь воздуха! Основы визуального дизайна для графики веб и мультимедиа / К. Голомбински, Р. Хаген; Пер. с англ. Н.А. Римицан.. - СПб.: Питер, 2013. - 272 с.

19. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. - М.: Эксмо, 2015. - 480 с.

20. Кирсанов Д. Веб-дизайн: книга Дмитрия Кирсанова / Д. Кирсанов. - М.: Символ, 2015. - 368 с.

21. Макнейл П. Настольная книга веб-дизайнера / П. Макнейл. - СПб.: Питер, 2013. - 264 с.

22. Нильсен, Я. Веб-дизайн: книга Якоба Нильсена / Я. Нильсен. - М.: Символ, 2015. - 512 с.

23. Седерхольм Д. Пуленепробиваемый веб-дизайн. Библиотека специалиста / Д. Седерхольм. - СПб.: Питер, 2012. - 304 с.

24. Седерхольм Д. Пуленепробиваемый веб-дизайн / Д. Седерхольм. - СПб.: Питер, 2012. - 304 с.

25. Фельке-Моррис Т. Большая книга веб-дизайна / Т. Фельке -Моррис. - М.: Эксмо, 2012. - 608 с.

26. Трофимов В.В. Информационные системы и технологии в экономике и управлении – М.:Высшее образование, 2007. – 480 с.

27. Ивасенко А.Г., Гридасов А.Ю., Павленко В.А. Информационные системы и технологии в экономике и управлении – М.: КноРус, 2007. – 160 с.

28. Михеева Е.В. Информационные технологии в профессиональной

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вым.	Арк.	№ докум.	Підпис	Дата		52

деятельности. – М.: Академия, 2007 – 379 с.

29. Грабауров В.А. Информационные технологии для менеджеров (Прикладные информационные технологии). М.; ИНФРА-М 2001. – 286 с.

30. Алиев В.С. Информационные технологии и системы финансового менеджмента. – М.: ИНФРА-М, Форум, 2007. – 320 с.

31. ERP системы. Современное планирование и управление ресурсами предприятия. М. : Вершина, 2004 г., 272 с.

32. Черкашин П. Стратегия управления взаимоотношениями с клиентами (CRM) – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007 – 384 с.

33. Тидвелл Дж. Разработка пользовательских интерфейсов.— Питер, 2007.

34. Мунипов В. М., Зинченко В. П. Эргономика.— Логос, 2001.

35. Дегтяренко И. А. Теория деятельности и разработка информационных систем, 2005.

36. Шенг Лиэнг Интерфейс JNI. Руководство по программированию и спецификация / Шенг Лиэнг. - М.: ДМК Пресс, 2016. - 849 с.

37. Басов К. Деятельностный подход при проектировании человеко-компьютерного взаимодействия. На примере медицинских интерфейсов / В.Л. Авербух и др. - М.: Ленанд, 2014. - 527 с.

38. Джесси Рассел Интерфейс программирования приложений / Джесси Рассел. - М.: VSD, 2012. - 811 с.

39. Купер Алан об интерфейсе. Основы проектирования взаимодействия / Алан Купер. - М.: Символ-плюс, 2006. - 765 с.

40. Дари К. AJAX и PHP. Разработка динамических веб-приложений / К. Дари, Б. Бринзаре и др. - М.: Символ, 2015. - 336 с.

41. Эспозито Д. Разработка современных веб-приложений: анализ предметных областей и технологий - М.: Вильямс И.Д., 2017. – 464 с.

42. Афанасьев А.А. Аутентификация. Теория и практика обеспечения

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

безопасного доступа к информационным ресурсам. Учебное пособие для вузов. Гриф УМО МО РФ: моногр. / Афанасьев Алексей Алексеевич. - М.: Горячая линия - Телеком, 2017. - 270 с.

43. Еременко А.В. Двухфакторная аутентификация пользователей компьютерных систем на удаленном сервере по клавиатурному почерку / А.В. Еременко. - М.: Синергия, 2015. - 130 с.

44. Рассел Дж. Аутентификация. - М.: VSD, 2014. - 473 с.

45. Рассел Дж. Единая система идентификации и аутентификации / Джесси Рассел. - М.: VSD, 2016. - 717 с.

46. Атре Ш. Структурный подход к организации баз данных. - М.: Финансы и статистика, 2010. - 317 с.

47. Редько, В.Н. Базы данных и информационные системы / В.Н. Редько, И.А. Бассараб. - М.: Знание, 2004. - 240 с.

48. Дейт К.Дж. Введение в системы баз данных / К.Дж. Дейт. - К.: Диалектика; Издание 6-е, 2004. - 784 с.

49. Ульман, Дж. Основы систем баз данных / Дж. Ульман. - М.: Финансы и статистика, 2017. - 292 с.

50. Белоногов Г.Г., Новоселов А.П. Автоматизация процессов накопления, поиска и обобщения информации - М.: Наука, 2017. - 256 с.

					КБР-123.21.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	6
9 Порядок контролю та приймання.....	6

КБР-123.21.0054.00.00.ТЗ

Вим.	Арк.	№ документа	Підпис	Дата			
Розробив		Кушніков І.О.			Літ.	Аркуш	Аркушів
Перевірів		Улічев О.С.					
Н. Контр.		Гермак В.С.			ЦНТУ КІ-19-2СК		
Затв.		Смірнов О.А.					

Програмне забезпечення системи
особистого онлайн організатору

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи особистого онлайн органайзеру.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 189-02 від 28.12.2020 року).

3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи особистого онлайн органайзеру.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					КБР-123.21.0054.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи особистого онлайн органайзеру;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					КБР-123.21.0054.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище PHP.

					КБР-123.21.0054.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 54 аркушів.

					КБР-123.21.0054.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання кваліфікаційної бакалаврської роботи на попередній захист 22.05.2021 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист 9.06.2021 р.

					КБР-123.21.0054.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи
_____ Улічев О.С.

Програмне забезпечення системи особистого онлайн організера

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 40

Літера: РП

Кропивницький – 2021 року

```
// index.php - головна сторінка веб-сайту
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Pandya | Зручне керування проектами</title>
    <link href="assets/css/libraries/bootstrap.min.css" rel="stylesheet">
    <link href="assets/css/index.css" rel="stylesheet">
    <link rel="shortcut icon" type="image/png" href="img/favicon.png"/>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12">
          <div class="logo">
            
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-md-12">
          <div class="heading">
            <h1>
              <span style="display: block; width: 100%; text-align:
center;"><span style="font-weight: 500">Pandya</span> вітає Вас!</span>
              <span style="font-weight: 500">Pandya</span> - Ваш цифровий
помічник. Просте та зручне управління бізнес-проектами. Швидкий та зручний
доступ до Ваших даних: <span style="font-style: italic">заміток, файлів,
документів</span>. А найголовніше - БЕЗКОШТОВНО.<br>
              <br><span class="text_center">Приєднуйтеся до нас!</span>
            </h1>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-md-12">
          <div class="login_buttons">
            <ul class="login_list">
              <li><a href="register" id="register_btn">Реєстрація</a></li>
              <li>або</li>
              <li><a href="login" id="login_btn">Увійти</a></li>
            </ul>
          </div>
        </div>
      </div>
    </div>
    <script src="assets/js/libraries/jquery-2.1.4.js"></script>
    <script src="assets/js/libraries/bootstrap.min.js"></script>
  </body>
</html>
```

//register.php - сторінка реєстрації

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Pandya | Зручне керування проектами</title>
    <link href="assets/css/libraries/bootstrap.min.css" rel="stylesheet">
    <link href="assets/css/register.css" rel="stylesheet">
    <link rel="shortcut icon" type="image/png" href="img/favicon.png"/>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12">
          <div class="logo">
            <a href="/"></a>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="registration_form">
          <h2 class="heading">Реєстрація</h2>
          <div class="error_notification">
            <span class="notification_message"></span>
          </div>
          <div class="form">
            <form method="post" action="/register" enctype= "multipart/form-
data" id="file_upload" name="file_upload" onsubmit="return validate();">
              <div class="user_photo_area">
                <label id="photo-label" for="photo_field">
                  Ваше фото
                </label>
                <span class="photo_message"></span>
                <span class="btn btn-file">
                  Обрати.. <input id="photo_field" name="file" type="file">
                </span>
                <input type="text" id="filename" name="filename" />
              </div>
              <div class="login_area">
                <label for="login_field">
                  Ваше ім'я
                </label>
                <span class="login_message"></span>
                <div class="field">
                  <input type="text" name="user_login" id="login_field"
autocomplete="off"/>
                </div>
              </div>
              <div class="email_area">
                <label for="email_field">
                  Ваш E-mail
                </label>
                <span class="email_message"></span>
                <div class="field">
                  <input type="email" name="user_email" id="email_field"
autocomplete="off"/>
                </div>
              </div>
              <div class="password_area">
                <label for="password_field">
                  Пароль
                </label>
                <span class="password_message"></span>
                <div class="field">
                  <input type="password" name="user_password"
id="password_field" autocomplete="off"/>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

        </div>
    </div>
    <div class="register_btn">
        <input type="submit" id="register" name="signup"
value="Зарегиструватись" />
    </div>
</form>
</div>
</div>
</div>
<script src="assets/js/libraries/jquery-2.1.4.js"></script>
<script src="assets/js/libraries/bootstrap.min.js"></script>
<script src="assets/js/register.js"></script>
</body>
</html>
<?php

session_start();

$content = json_decode(file_get_contents("config/language.json"));
$lang = $_SESSION['lang'];
if($lang == '' || $lang == 'ua'){
    $words = $content->language[0]->ua[0];
} else if($lang == 'uk'){
    $words = $content->language[1]->uk[0];
} else {
    $words = $content->language[2]->ru[0];
}

$config = include "config/db.php";

if (isset($_POST['signup'])) {
    $login = $_POST['user_login'];
    $password = password_hash($_POST['user_password'], PASSWORD_DEFAULT);
    $email = $_POST['user_email'];

    $file_exts = array("jpg", "jpeg", "png");
    $temp = explode(".", $_FILES["file"]["name"]);
    $upload_exts = end($temp);
    $file_name = md5(date('m/d/Y h:i:s a', time()) . $_FILES["file"]["name"]) .
"." . $upload_exts;
    if ((($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/png")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 8000000)
&& in_array($upload_exts, $file_exts)) {

        if ($_FILES["file"]["error"] > 0) {
            echo "<script> showError(\"Помилка! Код: " . $_FILES["file"]["error"] .
"\", false);</script>";
        } else {
            if (file_exists("/images/user_avatars/" . $file_name)) {
                echo "<script> showError(\"Файл з назвою " . $_FILES["file"]["name"] . "
вже існує.\", false);</script>";
            } else {
                move_uploaded_file($_FILES["file"]["tmp_name"], "images/user_avatars/" .
$file_name);
                try {
                    $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);
                    $sth = $dbh->prepare("INSERT INTO users (username, email, password,
active, photo) values ('$login', '$email', '$password', 0, '$file_name')");
                    $sth->execute();

                    $userid = $dbh->lastInsertId();

                    $key = $login . $email . date('Y-m-d H:i:s');
                    $key = md5($key);

```

```

        $sth = $dbh->prepare("INSERT INTO confirm VALUES (NULL, '$userid',
'$key', '$email', 0, 1)");
        $sth->execute();

        require("assets/class.phpmailer.php");

        $mail = new PHPMailer();
        $mail->CharSet = 'UTF-8';
        $mail->From = "noreply@pandya.esy.es";
        $mail->FromName = "Pandya";
        $mail->AddAddress($email);
        $mail->Subject = $words->email_subject;
        $mail->Body = $words->email_body .
"\n\nhttp://pandya.esy.es/confirm.php?username=" . $login . "&email=" . $email .
"&secure_key=" . $key;

        if(!$mail->Send()) {
            echo 'Mailer error: ' . $mail->ErrorInfo;
        } else {
            echo "<script>
                showError(\"Протягом хвилини на " . $email . " буде надіслано
листа з кодом підтвердження реєстрації!\", true);
                $(\"#login_field\").val('');
                $(\"#email_field\").val('');
                $(\"#password_field\").val('');
            </script>";
        }
    } catch (PDOException $e){
        echo $e->getMessage();
    }
    $dbh = null;
}
} else {
    echo "<script> showError(\"Недопустимий файл.\", false);</script>";
}
}
?>

```

//register.js - скрипт сторінки реєстрації

```

var can_register, fileinput;

$(function() {

    can_register = { login: true, email: true, pass: true, photo: true };

    $(document).on('change', '.btn-file :file', function() {
    fileinput = this,
    input = $(this),
    numFiles = input.get(0).files ? input.get(0).files.length : 1,
    label = input.val().replace(/\\/g, '/').replace(/.*\/$/, '');
    input.trigger('fileselect', [numFiles, label]);
    });

    function readURL(input) {
    if (input.files && input.files[0]) {
    var reader = new FileReader();
    reader.onload = function (e) {
    $('#logo-image').attr('src', e.target.result);
    }
    reader.readAsDataURL(input.files[0]);
    }
    }

    $('.btn-file :file').on('fileselect', function(event, numFiles, label) {
    var _validFileExtensions = [".jpg", ".jpeg", ".png"];
    var sFileName = $("#photo_field").val();
    if (sFileName.length > 0) {
    var blnValid = false;
    for (var j = 0; j < _validFileExtensions.length; j++) {
    var sCurExtension = _validFileExtensions[j];
    if (sFileName.substr(sFileName.length - sCurExtension.length,
    sCurExtension.length).toLowerCase() == sCurExtension.toLowerCase()) {
    blnValid = true;
    break;
    }
    }
    }
    $("#filename").val(label);
    if (!blnValid) {
    $("#photo-label").css({"display": "inline-block"});
    $(".photo_message").html("Файл не є зображенням!");
    $('#logo-image').attr('src',
    "assets/img/logo_short.png");
    can_register.photo = false;
    } else {
    $("#photo-label").css({"display": "block"});
    $(".photo_message").html("");
    $(".error_notification").css({"height": "0px", "background-color":
    "#fff", "border-left": "0px solid #fff"});
    $(".notification_message").html("");
    readURL(this);
    }
    }
    });

    $("#login_field").attr("maxlength", 40);
    $("#email_field").attr("maxlength", 40);
    $("#password_field").attr("maxlength", 40);

    $("#login_field").keyup(function(event) {
    var login = $("#login_field").val();

    if(login.length < 4){
    $(".login_message").html("Занадто коротке ім'я!");
    can_register.login = false;
    return false;
    }
    }
    });

```

```

if(login.length > 15){
    $(".login_message").html("Занадто довге ім'я!");
    can_register.login = false;
    return false;
}
if(!/^[a-zA-Z0-9_-]{3,15}$/.test(login)){
    $(".login_message").html("Некоректні символи в імені!");
    can_register.login = false;
    return false;
}

check('login');
if(event.keyCode == '13'){
    $("#register").click();
}
});

$("#email_field").keyup(function(event){
    var email = $("#email_field").val();

    if(!/^[^<>() [\]\.\.,;:\s@\"']+(\.[^<>() [\]\.\.,;:\s@\"']+)*|(\\".+\\")@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\)|((\[[a-zA-Z\ -0-9]+\.\.)+[a-zA-Z]{2,}))$/.test(email)){
        $(".email_message").html("Не дійсний E-mail!");
        can_register.email = false;
        return false;
    }

    check('email');
    if(event.keyCode == '13'){
        $("#register").click();
    }
});

$("#password_field").keyup(function(event){
    var pass = $("#password_field").val();

    if(pass.length < 6){
        $(".password_message").html("Занадто короткий пароль!");
        can_register.pass = false;
    } else {
        $(".password_message").html('');
        can_register.pass = true;
    }

    if(event.keyCode == '13'){
        $("#register").click();
    }
});

function check(field){
    var value = $("#" + field + "_field").val(),
        errors = {login: "Логін вже зайнятий!", email: "E-mail вже зайнятий!"};

    $.ajax({
        type: "POST",
        url: "ajax/register.php",
        data: {
            check : true,
            field : field,
            value : value
        }
    })
    .done(function(msg){
        if(msg == 'false'){
            $(". " + field + "_message").html(errors[field]);
            can_register[field] = false;

```

```
        } else {
            $("._" + field + "_message").html('');
            can_register[field] = true;
        }
    });
}

});

function showError(text, flag) {
    if (flag) {
        $(".error_notification").css({"height": "70px", "background-color":
"#D8FFD5", "border-left": "7px solid #78FF6E"});
    } else {
        $(".error_notification").css({"height": "50px", "background-color":
"#F9D8D3", "border-left": "7px solid #F9756B"});
    }
    $(".notification_message").html(text);
}

function validate() {
    var login = $("#login_field").val(),
        email = $("#email_field").val(),
        pass = $("#password_field").val(),
        photo = $("#photo_field").val();

    if(login == '' || email == '' || pass == '' || photo == '') {
        showError("Заповніть порожні поля!", false);
        return false;
    }
    if(!(can_register.login && can_register.email && can_register.pass &&
can_register.photo)) {
        showError("Перевірте заповнені поля на правильність!", false);
        return false;
    }
    $(".error_notification").css({"height": "0px"});
    return true;
}
```

// recover.php - сторінка відновлення паролю

```

<?php include 'blocks/get_lang.php'; ?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Pandya | Відновлення паролю</title>
    <link href="assets/css/libraries/bootstrap.min.css" rel="stylesheet">
    <link href="assets/css/recover.css" rel="stylesheet">
    <link rel="shortcut icon" type="image/png" href="img/favicon.png"/>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12">
          <div class="logo">
            <a href="/"></a>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="recovery">
          <?php if(empty($_GET['email']) || empty($_GET['secure_key'])): ?>
            <h2 class="heading"><?php echo $words->recover_title; ?></h2>
            <div class="error_notification">
              <span class="notification_message"></span>
            </div>
            <div class="form">
              <div class="email_area">
                <label for="email_field"><?php echo $words->recover_email_text;
?></label>
                <div class="field">
                  <input type="text" id="email_field" autocomplete="off"/>
                </div>
                <div class="recover_btn">
                  <input type="button" id="recover" value="<?php echo $words-
>recover_button_text; ?>" />
                </div>
              </div>
            <?php else:
              $config = include "config/db.php";

              $email = $_GET['email'];
              $key = $_GET['secure_key'];

              try{

                $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);

                $sth = $dbh->query("SELECT * FROM confirm WHERE email = '$email'
AND secure_key = '$key' AND change_pass = 1 LIMIT 1");
                $sth->setFetchMode(PDO::FETCH_ASSOC);

                $row = $sth->fetch();

                if($row['id'] == ''){ ?>
                  <h2 class='heading'><?php echo $words->recover_title; ?></h2>
                  <div class='form'>
                    <div class='error_notification' style='height: 50px;
background-color: #F9D8D3; border-left: 7px solid #F9756B; padding: 0'>
                      <span class='notification_message'><?php echo $words-
>confirm_url_error_text; ?></span>
                    </div>
                  </div>
                <?php } else { ?>

```

```

<h2 class='heading'><?php echo $words->recover_title; ?></h2>
<div class='error_notification'>
  <span class='notification_message'></span>
</div>
<div class='form'>
  <div class='new_password_area'>
    <label for='new_password_field'>
      <?php echo $words->recover_new_password; ?>
    </label>
    <span class='new_password_message'></span>
    <div class='field'>
      <input type='password' id='new_password_field'
autocomplete='off' />
    </div>
  </div>
  <div class='new_password2_area'>
    <label for='new_password2_field'>
      <?php echo $words->recover_new_password_confirm; ?>
    </label>
    <span class='new_password2_message'></span>
    <div class='field'>
      <input type='password' id='new_password2_field'
autocomplete='off' />
    </div>
  </div>
  <div class='recover_btn'>
    <input type='button' id='change_pass' value='<?php echo $words-
>recover_changepassbtn_text; ?>' />
  </div>
</div>
<?php }

} catch (PDOException $e) {
  echo $e->getMessage();
}

$dbh = null;
?>
<?php endif; ?>
      </div>
    </div>
  </div>
<script type="text/javascript">
var counter_title = "<?php echo $words->counter_title; ?>",
  register_empty_field = "<?php echo $words->register_empty_field; ?>",
  incorrect_login_or_password = "<?php echo $words-
>incorrect_login_or_password; ?>",
  register_empty_field = "<?php echo $words->register_empty_field; ?>",
  register_check_fields = "<?php echo $words->register_check_fields; ?>",
  register_minute_first = "<?php echo $words->register_minute_first; ?>",
  recover_minute_second = "<?php echo $words->recover_minute_second; ?>",
  register_not_valid_email = "<?php echo $words->register_not_valid_email;
?>",
  recover_email_not_exists = "<?php echo $words->recover_email_not_exists;
?>",
  register_short_password = "<?php echo $words->register_short_password;
?>",
  recover_password_dont_match = "<?php echo $words-
>recover_password_dont_match; ?>",
  recover_password_match = "<?php echo $words->recover_password_match; ?>",
  recover_password_changed = "<?php echo $words->recover_password_changed;
?>";
var email = <?php echo json_encode($_GET['email']); ?>;
</script>
<script src="assets/js/libraries/jquery-2.1.4.js"></script>
<script src="assets/js/libraries/bootstrap.min.js"></script>
<script src="assets/js/recover.js"></script>
</body>
</html>

```

// recover.js - скрипт сторінки відновлення паролю

```

$(document).ready(function(){
    var can_register = {email: true, pass: true, pass_length: true};

    $("#email_field").attr("maxlength", 40);
    $("#new_password_field").attr("maxlength", 40);
    $("#new_password2_field").attr("maxlength", 40);

    $("#recover").click(function(){
        var email = $("#email_field").val();

        if(email == '')
        {
            $(".error_notification").css({"height": "50px", "background-color": "#F9D8D3", "border-left": "7px solid #F9756B"});
            $(".notification_message").html(register_empty_field);
            return false;
        }

        if(!(can_register.email && can_register.pass)){
            $(".error_notification").css({"height": "50px", "background-color": "#F9D8D3", "border-left": "7px solid #F9756B"});
            $(".notification_message").html(register_check_fields);
            return false;
        }

        $(".error_notification").css({"height": "0px", "background-color": "#fff", "border-left": "0px solid #fff"});
        $(".notification_message").html("");

        $.ajax({
            type: "POST",
            url: "ajax/check_email.php",
            data: {
                recover : true,
                email: email
            }
        })
        .done(function(msg){
            if(msg == 'success'){
                $(".error_notification").css({"height": "70px", "background-color": "#D8FFD5", "border-left": "7px solid #78FF6E"});
                $(".notification_message").html(register_minute_first + email + recover_minute_second);
                $("#email_field").val('');
            }
        });

        $("#email_field").keyup(function(event){
            var email = $("#email_field").val();

            if(!/^[^<>() []\.\.,;:\s@\"']+(\.[^<>() []\.\.,;:\s@\"']+)*|(\\".+\"))@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\)|((\[[a-zA-Z\0-9]+\.\.)+[a-zA-Z]{2,}))\$/ .test(email)){
                $(".error_notification").css({"height": "50px", "background-color": "#F9D8D3", "border-left": "7px solid #F9756B"});
                $(".notification_message").html(register_not_valid_email);
                can_register.email = false;
                return false;
            }

            $(".error_notification").css({"height": "0px", "background-color": "#fff", "border-left": "0px solid #fff"});
            $(".notification_message").html("");
            can_register.email = true;
        }
    });
}

```

```

        check('email');
        if(event.keyCode == '13'){
            $("#recover").click();
        }
    });

    function check(field){
        var value = $("#" + field + "_field").val(),
            errors = {email: recover_email_not_exists};

        $.ajax({
            type: "POST",
            url: "ajax/check_email.php",
            data: {
                check_registered_email : true,
                field : field,
                value : value
            }
        }).done(function(msg){
            if(msg == 'false'){
                $(".error_notification").css({"height": "50px",
                "background-color": "#F9D8D3", "border-left": "7px solid #F9756B"});
                $(".notification_message").html(errors[field]);
                can_register[field] = false;
            } else {
                $(".error_notification").css({"height": "0px",
                "background-color": "#fff", "border-left": "0px solid #fff"});
                $(".notification_message").html('');
                can_register[field] = true;
            }
        });
    }

    $("#new_password_field").keyup(function(){
        var new_pass = $("#new_password_field").val();

        check_passwords();

        if(new_pass.length < 6){
            $(".new_password_message").html(register_short_password);
            can_register.pass_length = false;
        } else {
            $(".new_password_message").html('');
            can_register.pass_length = true;
        }

        if(event.keyCode == '13'){
            $("#change_pass").click();
        }
    });

    $("#new_password2_field").keyup(function(){
        var new_pass2 = $("#new_password2_field").val();

        check_passwords();

        if(new_pass2.length < 6){
            $(".new_password2_message").html(register_short_password);
            can_register.pass_length = false;
        } else {
            $(".new_password2_message").html('');
            can_register.pass_length = true;
        }

        if(event.keyCode == '13'){
            $("#change_pass").click();
        }
    });

```

```

$("#change_pass").click(function(){
    var new_pass = $("#new_password_field"),
        new_pass2 = $("#new_password2_field");

    if(new_pass.val() == '' || new_pass2.val() == ''){
        $(".error_notification").css({"height": "50px", "background-
color": "#F9D8D3", "border-left": "7px solid #F9756B"});
        $(".notification_message").html(register_empty_field);
        return false;
    }

    if(!(can_register.pass && can_register.pass_length)){
        $(".error_notification").css({"height": "50px", "background-
color": "#F9D8D3", "border-left": "7px solid #F9756B"});
        $(".notification_message").html(register_check_fields);
        return false;
    }

    $(".error_notification").css({"height": "0px", "background-color":
"#fff", "border-left": "0px solid #fff"});
    $(".notification_message").html("");
    $.ajax({
        type: "POST",
        url: "ajax/check_email.php",
        data: {
            change_pass : true,
            email: email,
            password: new_pass.val()
        }
    })
    .done(function(msg){
        if(msg == 'success'){
            $(".error_notification").css({"height": "70px",
"background-color": "#D8FFD5", "border-left": "7px solid #78FF6E"});

            $(".notification_message").html(recover_password_changed);
            $("#email_field").val('');
        }
    });
});

function check_passwords(){
    var new_pass = $("#new_password_field"),
        new_pass2 = $("#new_password2_field");

    if(new_pass.val() != new_pass2.val()){
        $(".error_notification").css({"height": "50px", "background-
color": "#F9D8D3", "border-left": "7px solid #F9756B"});
        $(".notification_message").html(recover_password_dont_match);
        can_register.pass = false;
    } else {
        $(".error_notification").css({"height": "70px", "background-
color": "#D8FFD5", "border-left": "7px solid #78FF6E"});
        $(".notification_message").html(recover_password_match);
        can_register.pass = true;
    }
}

});
}

```

// ajax/check_email.php - Обробник AJAX-запитів на перевірку існуючого e-mail'у або логіну

```

<?php

session_start();

$content = json_decode(file_get_contents("../config/language.json"));
$lang = $_SESSION['lang'];
if($lang == '' || $lang == 'ua'){
    $words = $content->language[0]->ua[0];
} else if($lang == 'uk'){
    $words = $content->language[1]->uk[0];
} else {
    $words = $content->language[2]->ru[0];
}

$config = include "../config/db.php";

if(isset($_POST['check_registered_email'])){
    $field = $_POST['field'];
    $value = $_POST['value'];

    try{

        $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);

        $sth = $dbh->query("SELECT email FROM users");

        $sth->setFetchMode(PDO::FETCH_ASSOC);

        while($row = $sth->fetch()){
            if($row['email'] === $value){
                echo 'true';
                exit();
            }
        }
        echo 'false';
    } catch (PDOException $e){
        echo $e->getMessage();
    }

    $dbh = null;
}

if(isset($_POST['recover'])){
    $email = $_POST['email'];

    try{

        $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);

        $sth = $dbh->query("SELECT id, username FROM users WHERE email =
'$email' ");

        $sth->setFetchMode(PDO::FETCH_ASSOC);

        $row = $sth->fetch();
        if($row['id'] != ''){
            $key = $row['id'] . $row['username'] . $email . date('Y-m-d
H:i:s');

            $key = md5($key);

            $sth = $dbh->prepare("INSERT INTO confirm VALUES (NULL,
'$row[id]', '$key', '$email', 1, 0)");
            $sth->execute();
        }
    }
}

```

```

require("../assets/class.phpmailer.php");

$mail = new PHPMailer();
$mail->CharSet = 'UTF-8';
$mail->From = "noreply@pandya.esy.es";
$mail->FromName = "Pandya";
$mail->AddAddress($email);
$mail->Subject = $words->recover_email_subject;
$mail->Body = $words->recover_email_body .
"\n\nhttp://pandya.esy.es/recover.php?email=" . $email . "&secure_key=" . $key;

if(!$mail->Send()) {
    echo 'Mailer error: ' . $mail->ErrorInfo;
} else {
    echo "success";
}
}
} catch (PDOException $e){
    echo $e->getMessage();
}

$dbh = null;
}

if(isset($_POST['change_pass'])){
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);

    try{

        $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);

        $sth = $dbh->query("DELETE FROM confirm WHERE email = '$email' ");

        $sth = $dbh->prepare("UPDATE users SET password = '$password' WHERE
email = '$email' ");
        $sth->execute();
        $sth->setFetchMode(PDO::FETCH_ASSOC);
        if($sth){
            echo "success";
        } else {
            echo "error";
        }
    } catch (PDOException $e){
        echo $e->getMessage();
    }
    $dbh = null;
}

```

// ajax/check_user.php - Обробник AJAX-запитів на перевірку логіну користувача на існуючий у БД

```

<?php
session_start();

$config = include "../config/db.php";

if(isset($_POST['check_user'])){

    $user = $_POST['username'];
    $pass = $_POST['password'];

    try{

        $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);

        $sth = $dbh->prepare("SELECT * FROM users WHERE username =
'$user'");
        $sth->execute();

        $sth->setFetchMode(PDO::FETCH_ASSOC);

        $row = $sth->fetch();

        if($row['id'] != ''){
            if (password_verify($pass, $row['password']) && $row['active']
== 1) {

                echo 'true';
                $_SESSION['is_logged'] = true;
                $_SESSION['login'] = $user;
                setcookie("login", $user, strtotime( '+30 days' ), "/",
"", "", TRUE);
                setcookie("id", $row['id'], strtotime( '+30 days' ),
"/", "", "", TRUE);
            }
            echo 'false';

        } catch (PDOException $e){
            echo $e->getMessage();
        }

        $dbh = null;
    }

?>

```

// confirm.php - сторінка підтвердження реєстрації

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Pandya | Зручне керування проектами</title>
    <link href="assets/css/libraries/bootstrap.min.css" rel="stylesheet">
    <link href="assets/css/confirm.css" rel="stylesheet">
    <link rel="shortcut icon" type="image/png" href="img/favicon.png"/>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12">
          <div class="logo">
            <a href="/"></a>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="confirmed">
          <h2 class="heading">Підтвердження реєстрації</h2>
          <?php

            $config = include "config/db.php";

            if(empty($_GET['email']) || empty($_GET['secure_key'])){
              echo "<div class='error_notification' style='height: 50px;
background-color: #F9D8D3; border-left: 7px solid #F9756B'>
              <span class='notification_message'>Сталася помилка, будь ласка
перевірте свій E-mail ще раз!</span>
              </div>";
            } else {

              $username = $_GET['username'];
              $email = $_GET['email'];
              $key = $_GET['secure_key'];

              try{

                $dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);

                $sth = $dbh->query("SELECT * FROM confirm WHERE email = '$email'
AND secure_key = '$key' AND reg_confirm = 1 LIMIT 1");
                $sth->setFetchMode(PDO::FETCH_ASSOC);

                $row = $sth->fetch();

                if($row['id'] == ''){
                  echo "<div class='error_notification' style='height: 50px;
background-color: #F9D8D3; border-left: 7px solid #F9756B'>
                  <span class='notification_message'>Дане посилання більше
недійсне!</span>
                  </div>";
                } else {

                  $sth = $dbh->prepare("UPDATE users SET active = 1 WHERE id =
'$row[user_id]' LIMIT 1 ");
                  $sth->execute();

                  $dbh->exec("DELETE FROM confirm WHERE id = '$row[id]' LIMIT 1
");

                  $sth = $dbh->prepare("INSERT INTO boards VALUES (NULL,
'Привітальна дошка', '#19AA00', 'public')");
                  $sth->execute();

```

```

        $lastId = $dbh->lastInsertId();
        $dbh->exec("INSERT INTO user_boards VALUES (NULL,
'$row[user_id]', '$lastId', '1', '', '0')");
        $dbh->exec("INSERT INTO lists VALUES (NULL, 'Зразковий список',
'$lastId', '0')");
        $listID = $dbh->lastInsertId();
        $dbh->exec("INSERT INTO tasks VALUES (NULL, '$listID', 'Зразкове
завдання', '', '', '', '')");

        if($dbh){
            echo "<div class='error_notification' style='height: 50px;
background-color: #D8FFD5; border-left: 7px solid #78FF6E'>
                <span class='notification_message'>Аккаунт <span
style='font-weight: 400'>" . $username . "</span> активовано!</span>
            </div>";

            echo "<div> <a href=\"/login\"
id=\"login_btn\">Авторизація</a></div>";
        }
    } catch (PDOException $e){
        echo $e->getMessage();
    }

    $dbh = null;
}

?>
</div>
</div>
</div>
<script src="assets/js/libraries/jquery-2.1.4.js"></script>
<script src="assets/js/libraries/bootstrap.min.js"></script>
</body>
</html>

```

// login.php - сторінка авторизації

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Авторизація</title>
    <link href="assets/css/libraries/bootstrap.min.css" rel="stylesheet">
    <link href="assets/css/login.css" rel="stylesheet">
    <link rel="shortcut icon" type="image/png" href="img/favicon.png"/>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12">
          <div class="logo">
            <a href="/"></a>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-md-12">
          <div class="login_form">
            <h2 class="heading">Авторизація</h2>
            <div class='login_error_confirm'>
              <span class='login_confirm_message'></span>
            </div>
            <div class="username_field">
              <label for="login_username">Ваше Ім'я</label>
              <div class="field">
                <input type="text" id="login_username" />
              </div>
            </div>
            <div class="pass_field">
              <label for="login_password">Пароль</label>
              <div class="field">
                <input type="password" id="login_password"/>
              </div>
            </div>
            <div class="login_form_buttons">
              <div class="clearfix">
                <input type="button" id="login_signup" value="Увійти"/>
                <a href="recover" class="forgot_pass">Забули пароль?</a>
              </div>
              <a href="register" class="forgot_pass_reg">Реєстрація</a>
            </div>
          </div>
        </div>
      </div>
      <script src="assets/js/libraries/jquery-2.1.4.js"></script>
      <script src="assets/js/libraries/bootstrap.min.js"></script>
      <script src="assets/js/login.js"></script>
    </body>
  </html>

```

```
// login.js - скрипт сторінки авторизації
```

```
$(function(){  
  
  $("#login_username").keyup(function(event){  
    if(event.keyCode == '13'){  
      $("#login_signup").click();  
    }  
  });  
  
  $("#login_password").keyup(function(event){  
    if(event.keyCode == '13'){  
      $("#login_signup").click();  
    }  
  });  
  
  $("#login_signup").click(function(){  
    var username = $("#login_username").val(),  
        password = $("#login_password").val();  
  
    if(username == '' || password == ''){  
      $('.login_error_confirm').css({'height': '35px', 'background-color':  
'#F9D8D3', 'border-left': '6px solid #F9756B'});  
      $('.login_confirm_message').html("Заповніть порожні поля!");  
      return false;  
    } else {  
      $('.login_error_confirm').css({'height': '0px', 'background-color':  
'#fff', 'border-left': '0px', 'margin': '0 auto'});  
      $('.login_confirm_message').html('');  
    }  
  
    $.ajax({  
      type: "POST",  
      url: "ajax/check_user.php",  
      data: {  
        check_user: true,  
        username: username,  
        password: password  
      }  
    }).success(function(msg){  
      if(msg == 'false'){  
        $('.login_error_confirm').css({'height': '50px', 'background-color':  
'#F9D8D3', 'border-left': '6px solid #F9756B'});  
        $('.login_confirm_message').html("Неправильний логін або пароль!");  
      } else {  
        window.location.href = "/me";  
      }  
    });  
  });  
});
```

```
// me.php - сторінка записів до органайзеру
```

```
<?php

session_start();

if($_SESSION['is_logged_in']){
    $login = $_SESSION['login'];
} else if($_COOKIE['login'] != ''){
    $login = $_COOKIE['login'];
}

if(!$login) {
    header( 'Location: /login', true, 307);
}

$config = include "config/db.php";
require 'assets/carbon/Carbon.php';
use Carbon\Carbon;

$user_id = $_COOKIE['id'];
$dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
    $config['username'], $config['password']);
$stmt = $dbh->prepare("SELECT * FROM users WHERE id=:user_id");
$stmt->execute(array('user_id' => $user_id));
$row = $stmt->fetch(PDO::FETCH_ASSOC);

$photo = $row['photo'];
$username = $row['username'];

function hex2rgb($hex) {
    $hex = str_replace("#", "", $hex);
    if(strlen($hex) == 3) {
        $r = hexdec(substr($hex,0,1).substr($hex,0,1));
        $g = hexdec(substr($hex,1,1).substr($hex,1,1));
        $b = hexdec(substr($hex,2,1).substr($hex,2,1));
    } else {
        $r = hexdec(substr($hex,0,2));
        $g = hexdec(substr($hex,2,2));
        $b = hexdec(substr($hex,4,2));
    }
    $rgb = array($r, $g, $b);
    return $rgb;
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Головна сторінка | Pandya</title>
    <link rel="stylesheet" href="assets/css/libraries/bootstrap.min.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
    <link href="assets/css/me.css" rel="stylesheet">
    <link rel="shortcut icon" type="image/png" href="assets/img/favicon.png"/>
</head>
<body>
    <div class="container-fluid">
        <?php include 'blocks/header_line.php' ?>
        <div class="main_container row">
            <div class="boards_container">
                <?php
                    $stmt = $dbh->prepare("SELECT board_id, last_seen, starred FROM
user_boards WHERE user_id=:user_id");
                    $stmt->execute(array('user_id' => $user_id));
                    $boards = array();
```

```

while($row = $sth->fetch(PDO::FETCH_ASSOC)) {
    array_push($boards, array("id" => $row['board_id'], "last_seen" =>
$row['last_seen'], "starred" => $row['starred']));
}
$user_board_ids = array();
$starred_boards_ids = array();
foreach ($boards as $board) {
    array_push($user_board_ids, $board["id"]);
    $starred_boards_ids[$board["id"]] = $board["starred"];
}
$user_board_ids = join(',', $user_board_ids);
$sth = $dbh->query("SELECT * FROM boards WHERE id IN
($user_board_ids) ORDER BY id DESC");
$all_boards = array();
$starred_boards = array();
while($row = $sth->fetch(PDO::FETCH_ASSOC)) {
    if($starred_boards_ids[$row['id']] == "1") {
        array_push($starred_boards, $row);
    }
    array_push($all_boards, $row);
}

$starred_boards_display_value = count($starred_boards) > 0 ?
"inline-block" : "none";
echo '<div class="starred_boards" style="display: ' .
$starred_boards_display_value . '">
    <div class="boards_heading">
        <i class="boards_heading_icon fa fa-star-o"> <span
class="boards_heading_text">Важливі дошки</span></i>
    </div>';
if(count($starred_boards) > 0) {
    foreach ($starred_boards as $board) {
        echo '<div class="col-sm-3">
            <div class="board" id="' . $board['id'] . '">
                <a class="open_board" href="/board?show=' . $board['id'] .
'" style="background-color: ' . $board['background'] . ';">
                    <span class="board_details">
                        <span class="board_title">' . $board['name'] .
'</span>
                            </span>
                        <i class="star_icon board_starred fa fa-star"></i>
                    </a>
                </div>
            </div>';
    }
}
echo '</div>';

echo '<div class="all_boards">
    <div class="boards_heading">
        <i class="boards_heading_icon fa fa-clone"> <span
class="boards_heading_text">Мої дошки</span></i>
    </div>';
    foreach ($all_boards as $board) {
        echo '<div class="col-sm-3">
            <div class="board" id="' . $board['id'] . '">
                <a class="open_board" href="/board?show=' . $board['id'] .
'" style="background-color: ' . $board['background'] . ';">
                    <span class="board_details">
                        <span class="board_title">' . $board['name'] . '</span>
                    </span>
                    <i class="star_icon fa ' .
                    echo $starred_boards_ids[$board['id']] == 1 ? 'fa-star
board_starred ' : 'fa-star-o';
                    echo '></i>
                </a>
            </div>
        </div>';
    }
}

```

```

        echo '<div class="col-sm-3">
            <div class="board">
                <a class="create_board" href="#">
                    <span class="board_details">
                        <span class="board_title">Створити нову дошку..</span>
                    </span>
                </a>
            </div>
        </div>;
    echo "</div>";
    ?>
</div>
</div>
<div class="popup_boards">
    <div class="popup_find_boards">
        <input type="text" id="find_boards" placeholder="Знайти дошки по
імені...">
    </div>
    <?php
        $starred_display_value = count($starred_boards) > 0 ? "block" : "none";
        echo '<div class="popup_starred_boards" style="display: ' .
$starred_display_value . ';">
            <span class="popup_list_header"><i class="popup_list_icon fa fa-
star-o"></i> Важливі дошки</span>
            <span class="toggle_board_list"><i class="minus_icon fa fa-
minus"></i></span>
            <ul class="popup_boards_list">;
                if(count($starred_boards) > 0) {
                    foreach ($starred_boards as $board) {
                        $rgba = hex2rgb($board['background']);
                        $rgba = "rgba(" . $rgba[0] . ", " . $rgba[1] . ", " . $rgba[2] . ",
0.15);";
                        echo '<li class="popup_boards_list_item" id="' . $board['id'] . '"
style="background: ' . $rgba . '">
                            <a href="/board?show=' . $board['id'] . '" class="list_item_link
starred" style="border-left: 40px solid ' . $board['background'] . ';">' .
$board['name'] . '
                                <div class="list_item_icons icons_visible">
                                    <i class="star fa fa-star"></i>
                                </div>
                            </a>
                        </li>;
                    }
                }
            echo '</ul></div>;

            $recent_boards_index = array();
            for ($i = 0; $i < count($boards); $i++) {
                $board_datetime = Carbon::createFromFormat('Y-m-d H:i:s',
$boards[$i]["last_seen"], 'Europe/Kiev');
                $now = Carbon::now('Europe/Kiev');
                $hours = $board_datetime->diffInHours($now);
                if($hours == 0) {
                    array_push($recent_boards_index, $i);
                }
            }
            $has_recent_unstarred = false;
            for($i = 0; $i < count($recent_boards_index); $i++) {
                if($starred_boards_ids[$boards[$recent_boards_index[$i]]['id']] != 1)
            {
                $has_recent_unstarred = true;
            }
        }
        $recent_display_value = $has_recent_unstarred ? "block" : "none";
        echo '<div class="popup_recent_boards" style="display: ' .
$recent_display_value . ';">
            <span class="popup_list_header"><i class="popup_list_icon fa fa-
clock-o"></i> Останні дошки</span>

```

```

        <span class="toggle_board_list"><i class="minus_icon fa fa-
minus"></i></span>
        <ul class="popup_boards_list">';
        if(count($recent_boards_index) > 0) {
            for($i = 0; $i < count($recent_boards_index); $i++) {
                if($starred_boards_ids[$boards[$recent_boards_index[$i]]['id']] !=
1) {
                    $rgba =
hex2rgb($all_boards[$recent_boards_index[$i]]['background']);
                    $rgba = "rgba(" . $rgba[0] . ", " . $rgba[1] . ", " . $rgba[2] .
", 0.15);";
                    echo '<li class="popup_boards_list_item" id="' .
$boards[$recent_boards_index[$i]]['id'] . '" style="background: ' . $rgba . '">
                        <a href="/board?show=' .
$boards[$recent_boards_index[$i]]['id'] . '" class="list_item_link"
style="border-left: 40px solid ' .
$all_boards[$recent_boards_index[$i]]['background'] . ';">' .
$all_boards[$recent_boards_index[$i]]['name'] . '
                            <div class="list_item_icons">
                                <i class="remove_board fa fa-times"></i>
                                <i class="star fa fa-star-o"></i>
                            </div>
                        </a>
                    </li>';
                }
            }
        }
        echo '</ul>
        </div>';

        echo '<div class="popup_my_boards">
            <span class="popup_list_header"><i class="popup_list_icon fa fa-
clone"></i> Мої дошки</span>
            <span class="toggle_board_list"><i class="minus_icon fa fa-
minus"></i></span>
            <ul class="popup_boards_list">';
            foreach ($all_boards as $board) {
                $link_style = $starred_boards_ids[$board['id']] == 1 ? "starred" :
"slidable";
                $list_item_icons_style = $starred_boards_ids[$board['id']] == 1 ?
'icons_visible"><i class="star fa fa-star"></i>' : '<i class="star fa fa-star-
o"></i>';
                $rgba = hex2rgb($board['background']);
                $rgba = "rgba(" . $rgba[0] . ", " . $rgba[1] . ", " . $rgba[2] . ",
0.15);";
                echo '<li class="popup_boards_list_item" id="' . $board['id'] . '"
style="background: ' . $rgba . '">
                    <a href="/board?show=' . $board['id'] . '" class="list_item_link '
. $link_style . '" style="border-left: 40px solid ' . $board['background'] .
';">' . $board['name'] . '
                        <div class="list_item_icons ' . $list_item_icons_style . '</div>
                    </a>
                </li>';
            }
            echo '</ul>
            </div>';
        ?>
        <div class="add_board">
            <a href="#" id="popup_create_board">Створити нову дошку..</a>
        </div>
    </div>
    </div>
    <script type="text/javascript" src="assets/js/libraries/jquery-
2.1.4.js"></script>
    <script type="text/javascript"
src="assets/js/libraries/bootstrap.min.js"></script>
    <script type="text/javascript"
src="assets/js/libraries/bootbox.min.js"></script>
    <script type="text/javascript" src="assets/js/header_line.js"></script>

```

```
<script type="text/javascript" src="assets/js/me.js"></script>  
</body>  
</html>
```

Кафедра КБПЗ – 2021 рік

```
// me.js - скрипт сторінки проєктів користувача
```

```
$(document).ready(function() {

    $(document).on('click', '.star_icon', function(event) {
        if(event.target != this) { return false; }
        event = event || window.event;
        event.preventDefault();
        var id = $(this).parent().parent().attr('id');
        if($(this).hasClass('board_starred')){
            $('.starred_boards').find("#" + id).parent().remove();
            if($('.starred_boards').find($('.col-sm-3')).length == 0) {
                $('.starred_boards').css({'display' : 'none'});
            }
            $('.all_boards').find("#" +
id).find($(".star_icon")).removeClass('board_starred');
            $('.all_boards').find("#" + id).find($(".star_icon")).removeClass('fa-
star');
            $('.all_boards').find("#" + id).find($(".star_icon")).addClass('fa-
star-o');

            starredInPopup(id, $(".popup_my_boards .popup_boards_list").find("li#" +
id).find("i"));
        } else {
            $(this).removeClass('fa-star-o');
            $(this).addClass('fa-star');
            $(this).addClass('board_starred');
            if($('.starred_boards').find($('.col-sm-3')).length == 0) {
                $('.starred_boards').css({'display' : 'inline-block'});
            }
            $('.starred_boards .boards_heading').after('\
<div class="col-sm-3">\
    <div class="board" id="' + id + '">\
        <a class="open_board" href="/board?show=' + id + '"
style="background-color: ' + $(this).parent().css('background-color') + '">\
            <span class="board_details">\
                <span class="board_title">' +
$(this).parent().find($(".board_details")).html() + '</span>\
            </span>\
            <i class="star_icon board_starred fa fa-star"></i>\
        </a>\
    </div>\
</div>\
');

            starredInPopup(id, $(".popup_my_boards .popup_boards_list").find("li#" +
id).find("i"));
        }
    });

    function starredInMain(id, icon) {
        var id = icon.parent().parent().attr('id');
        if(icon.hasClass('board_starred')){
            $('.starred_boards').find("#" + id).parent().remove();
            if($('.starred_boards').find($('.col-sm-3')).length == 0) {
                $('.starred_boards').css({'display' : 'none'});
            }
            $('.all_boards').find("#" +
id).find($(".star_icon")).removeClass('board_starred');
            $('.all_boards').find("#" + id).find($(".star_icon")).removeClass('fa-
star');
            $('.all_boards').find("#" + id).find($(".star_icon")).addClass('fa-
star-o');
        } else {
            icon.removeClass('fa-star-o');
            icon.addClass('fa-star');
            icon.addClass('board_starred');
            if($('.starred_boards').find($('.col-sm-3')).length == 0) {
                $('.starred_boards').css({'display' : 'inline-block'});
            }
        }
    }
});
```

```

    }
    $(' .starred_boards .boards_heading').after('\
    <div class="col-sm-3">\
      <div class="board" id="' + id + '">\
        <a class="open_board" href="/board?show=' + id + '"
style="background-color: ' + icon.parent().css('background-color') + '">\
          <span class="board_details">\
            <span class="board_title">' +
icon.parent().find($(".board_details")).html() + '</span>\
          </span>\
          <i class="star_icon board_starred fa fa-star"></i>\
        </a>\
      </div>\
    </div>\
  ');
}
}

function starredInPopup(id, icon) {
  var link = icon.parents(':eq(1)'),
  starredBoards_ul = $(' .popup_starred_boards .popup_boards_list'),
  starredBoards_li = $(' .popup_starred_boards
.popup_boards_list').find("li#" + id),
  recentBoards_ul = $(' .popup_recent_boards .popup_boards_list'),
  li = icon.parents(':eq(2)');
  if(link.hasClass('starred')) {
    link.removeClass('starred');
    link.addClass('slidable');
    icon.removeClass("fa-star");
    icon.addClass("fa-star-o");
    icon.parent().removeClass("icons_visible");
    starredBoards_li.remove();
    if(starredBoards_ul.find($('li')).length == 0) {
      $(' .popup_starred_boards').css({'display' : 'none'});
    }

    $.when(check_if_recent(id)).done(function(msg) {
      if(msg) {
        if(recentBoards_ul.find($('li')).length == 0) {
          $(' .popup_recent_boards').css({'display' : 'block'});
        }
        recentBoards_ul.prepend('\
          <li class="popup_boards_list_item" id="' + id + '"
style="background: ' + li.css('background') + '">\
            <a href="/board?show=' + id + '" class="list_item_link"
style="border-left: ' + li.find('a').css('border-left') + ';">' +
li.find('a').text() + '\
              <div class="list_item_icons">\
                <i class="remove_board fa fa-times"></i>\
                <i class="star fa fa-star-o"></i>\
              </div>\
            </a>\
          </li>'
        );
      }
    });
    set_starred(id, "0");
  } else if(link.hasClass('slidable')) {
    link.removeClass('slidable');
    link.addClass('starred');
    icon.addClass("fa-star");
    icon.removeClass("fa-star-o");
    icon.parent().addClass("icons_visible");
    if(recentBoards_ul.find("li#" + id).length != 0) {
      recentBoards_ul.find("li#" + id).remove();
      if(recentBoards_ul.find($('li')).length == 0) {
        $(' .popup_recent_boards').css({'display' : 'none'});
      }
    }
  }
}

```

```

        if(starredBoards_ul.find($('li')).length == 0) {
            $('popup_starred_boards').css({'display' : 'block'});
        }
        starredBoards_ul.prepend('\
            <li class="popup_boards_list_item" id="' + id + '" style="background:' +
link.parent().css('background') + ';">\
                <a href="/board?show=' + id + '" class="list_item_link starred"
style="border-left: ' + link.css('border-left') + ';">' +
                    link.text() + '<div class="list_item_icons icons_visible">\
                        <i class="star fa fa-star"></i>\
                    </div>\
                </a>\
            </li>'
        );
        set_starred(id, "1");
    }
}
$(document).on('click', '.star', function(event) {
    if(event.target != this) { return false; }
    event = event || window.event;
    event.preventDefault();
    var parent = $(this).parents(':eq(4)');
    var list_item_id = $(this).parents(':eq(2)').attr('id');
    if(parent.hasClass("popup_starred_boards")) {
        var ul = $('popup_starred_boards .popup_boards_list'),
            li = ul.find("li#" + list_item_id);
        li.remove();
        if(ul.find($('li')).length == 0) {
            $('popup_starred_boards').css({'display' : 'none'});
        }
        var myBoards_li = $('popup_my_boards .popup_boards_list').find("li#" +
list_item_id);
        myBoards_li.find(".star").removeClass("fa-star");
        myBoards_li.find(".star").addClass("fa-star-o");
        myBoards_li.find(".list_item_icons").removeClass("icons_visible");
        myBoards_li.find("a").removeClass("starred");
        myBoards_li.find("a").addClass("slidable");

        $.when(check_if_recent(list_item_id)).done(function(msg) {
            if(msg) {
                if($('popup_recent_boards .popup_boards_list').find($('li')).length
== 0) {
                    $('popup_recent_boards').css({'display' : 'block'});
                }
                $('popup_recent_boards .popup_boards_list').prepend('\
                    <li class="popup_boards_list_item" id="' + list_item_id + '"
style="background: ' + li.css('background') + '">\
                        <a href="/board?show=' + list_item_id + '" class="list_item_link"
style="border-left: ' + li.find('a').css('border-left') + ';">' +
                            li.find('a').text() + '\
                            <div class="list_item_icons">\
                                <i class="remove_board fa fa-times"></i>\
                                <i class="star fa fa-star-o"></i>\
                            </div>\
                        </a>\
                    </li>'
                );
            }
        });
        starredInMain(list_item_id, $(".starred_boards").find("div#" +
list_item_id).find("i"));
        set_starred(list_item_id, "0");
    } else if(parent.hasClass("popup_my_boards")) {
        starredInPopup(list_item_id, $(this));
        starredInMain(list_item_id, $(".all_boards").find("div#" +
list_item_id).find("i"));
    } else if(parent.hasClass("popup_recent_boards")) {
        var link = $(this).parents(':eq(1)'),
            starredBoards_ul = $('popup_starred_boards .popup_boards_list'),

```

```

        recentBoards_ul = $('.popup_recent_boards .popup_boards_list'),
        recentBoards_li = $('.popup_recent_boards
.popup_boards_list').find("li#" + list_item_id);
        if(starredBoards_ul.find($('li')).length == 0) {
            $('.popup_starred_boards').css({'display' : 'block'});
        }
        starredBoards_ul.prepend('\
            <li class="popup_boards_list_item" id="' + list_item_id + '"
style="background:' + link.parent().css('background') + ';">\
                <a href="/board?show=' + list_item_id + '" class="list_item_link
starred" style="border-left: ' + link.css('border-left') + ';"> +
                    link.text() + '<div class="list_item_icons icons_visible">\
                        <i class="star fa fa-star"></i>\
                    </div>\
                </a>\
            </li>'
        );
        recentBoards_li.remove();
        if(recentBoards_ul.find($('li')).length == 0) {
            $('.popup_recent_boards').css({'display' : 'none'});
        }
        var myBoards_li = $('.popup_my_boards .popup_boards_list').find("li#" +
list_item_id);
        myBoards_li.find(".star").addClass("fa-star");
        myBoards_li.find(".star").removeClass("fa-star-o");
        myBoards_li.find(".list_item_icons").addClass("icons_visible");
        myBoards_li.find("a").addClass("starred");
        myBoards_li.find("a").removeClass("slidable");
        starredInMain(list_item_id, $(".all_boards").find("div#" +
list_item_id).find("i"));

        set_starred(list_item_id, "1");
    }
});

$(document).on('click', '.remove_board', function(event) {
    event = event || window.event;
    event.preventDefault();
    var ul = $(this).parents(':eq(3)'),
        li = $(this).parents(':eq(2)'),
        list_item_id = li.attr('id');
    li.remove();
    if(ul.find($('li')).length == 0) {
        $('.popup_recent_boards').css({'display' : 'none'});
    }

    $.ajax({
        method: "POST",
        url: "ajax/me.php",
        data: { remove_datetime: true, id: list_item_id }
    });
});

function check_if_recent(id) {
    return $.ajax({
        method: "POST",
        url: "ajax/me.php",
        data: { isrecent: true, id: id }
    }).done(function(msg) { });
}

function set_starred(id, value) {
    $.ajax({
        method: "POST",
        url: "ajax/me.php",
        data: { starred: true, id: id, value: value }
    });
}
});

```

// ажах/ме.php - обробник AJAX-запитів сторінки записів до органайзеру

```

<?php

$config = include "../config/db.php";
$dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);
require '../assets/carbon/Carbon.php';
use Carbon\Carbon;

if(isset($_POST['create_board'])) {
    $user_id = $_COOKIE['id'];
    $board_name = $_POST['name'];
    $sth = $dbh->query("INSERT INTO boards VALUES (NULL, '$board_name',
'#19AA00', 'public')");
    $lastId = $dbh->lastInsertId();
    $sth = $dbh->query("INSERT INTO user_boards VALUES (NULL, '$user_id',
'$lastId', '1', '', '0')");
    if($sth) {
        echo $lastId;
    } else {
        echo 'failed';
    }
}

if(isset($_POST['starred'])) {
    $board_id = $_POST['id'];
    $user_id = $_COOKIE['id'];
    $value = $_POST['value'];
    $sth = $dbh->query("UPDATE user_boards SET starred='$value' WHERE
board_id='$board_id' AND user_id='$user_id'");
    if($sth) {
        echo 'success';
    } else {
        echo 'failed';
    }
}

if(isset($_POST['remove_datetime'])) {
    $board_id = $_POST['id'];
    $user_id = $_COOKIE['id'];
    $sth = $dbh->query("UPDATE user_boards SET last_seen='' WHERE
board_id='$board_id' AND user_id='$user_id'");
    if($sth) { echo 'success'; } else { echo 'failed'; }
}

if(isset($_POST['isrecent'])) {
    $board_id = $_POST['id'];
    $user_id = $_COOKIE['id'];
    $sth = $dbh->query("SELECT last_seen FROM user_boards WHERE
board_id='$board_id' AND user_id='$user_id'");
    $row = $sth->fetch(PDO::FETCH_ASSOC);
    $board_datetime = Carbon::createFromFormat('Y-m-d H:i:s',
$row["last_seen"], 'Europe/Kiev');
    $now = Carbon::now('Europe/Kiev');
    $hours = $board_datetime->diffInHours($now);
    if($hours == 0) {
        echo true;
    } else {
        echo false;
    }
}

?>

```

// board.php - сторінка окремого запису

```

<?php

session_start();

$config = include "config/db.php";
require 'assets/carbon/Carbon.php';
require 'blocks/functions.php';
use Carbon\Carbon;

if($_SESSION['is_logged_in']){
    $login = $_SESSION['login'];
} else if($_COOKIE['login'] != ''){
    $login = $_COOKIE['login'];
}

if(!$login) {
    header( 'Location: /login', true, 307);
}

$user_id = $_COOKIE['id'];
$board_id = $_GET['show'];
$dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);
$stmt = $dbh->prepare("SELECT * FROM user_boards WHERE user_id=:user_id AND
board_id=:board_id");
$stmt->execute(array('user_id' => $user_id, 'board_id' => $board_id));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
$has_rows = $row['id'];
$board_starred = $row['starred'];
$user_permission = $row['permission'];

$stmt = $dbh->prepare("SELECT * FROM boards WHERE id=:board_id");
$stmt->execute(array('board_id' => $board_id));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
$board_name = $row['name'];
$background_color = $row['background'];
$board_type = $row['type'];

$stmt = $dbh->prepare("SELECT * FROM users WHERE id=:user_id");
$stmt->execute(array('user_id' => $user_id));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
$photo = $row['photo'];
$username = $row['username'];

?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title><? echo $board_name; ?> | Pandya</title>
<link rel="stylesheet" href="assets/css/libraries/bootstrap.min.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css">
<link href="assets/css/board.css" rel="stylesheet">
<link rel="stylesheet" type="text/css" href="assets/css/libraries/jquery-
ui.min.css">
<link rel="shortcut icon" type="image/png" href="assets/img/favicon.png"/>
</head>
<body style="background-color: <?php echo $background_color; ?>;">
<div class="container-fluid">
<?php include 'blocks/header_line.php' ?>
<div class="main_container row">
<?php if ($has_rows == "" && $board_type == "private"): ?>
<div class="error_user">

```

```

        <span>Дошка являється приватною, Ви не маєте повноважень
переглядати її контент!</span>
    </div>
    <?php else: ?>
    <div class="board_line">
        <div class="board_name">
            <span class="name"><? echo $board_name; ?></span>
        </div>
        <div class="board_state">
            <?php if ($board_starred): ?>
            <a href="#" class="board_star" style="background-
color: rgba(0, 0, 0, 0.15);">
                <i class="fa fa-star">
            <?php else: ?>
            <a href="#" class="board_star">
                <i class="fa fa-star-o">
            <?php endif; ?>
            </i>
        </a>
    </div>
    <div class="board_type">
        <a href="#">
            <?php if ($board_type == "private"): ?>
            <i class="fa fa-lock"></i> Приватна
            <?php else: ?>
            <i class="fa fa-lock"></i> Публічна
            <?php endif; ?>
        </a>
    </div>
</div>
<div class="content">
    <div class="margined row">
        <?php
            $sth = $dbh->prepare("SELECT * FROM lists
WHERE board_id=:board_id ORDER BY position ASC");
            $sth->execute(array('board_id' =>
$board_id));
            while($row = $sth-
>fetch(PDO::FETCH_ASSOC)) {
                echo '
                    <div
class="list_wrapper col-md-3" position="' . $row['position'] . '">
                    <div class="board_list"
                        <div
class="list_header">
                            <span>' .
$row['name'] . '</span>
                        </div>
                    <div
class="list_content">
                        ';
                        $row_sth = $dbh-
>prepare("SELECT * FROM tasks WHERE list_id=:list_id
ORDER BY id ASC");
                        $row_sth-
>execute(array('list_id' => $row['id']));
                        while($sub_row = $row_sth-
>fetch(PDO::FETCH_ASSOC)) {
                            echo '
                                <div
class="list_task" id="' . $sub_row['id'] . '">
                                    <span>' .
$sub_row['name'] . '</span>
                                </div>
                            ';
                        }
                    echo '
                </div>
            }
        </div>
    </div>

```



```

        foreach ($starred_boards as $board) {
            $rgba = hex2rgb($board['background']);
            $rgba = "rgba(" . $rgba[0] . ", " . $rgba[1] . ", " . $rgba[2]
. ", 0.15);";
            echo '<li class="popup_boards_list_item" id="' . $board['id']
. '" style="background: ' . $rgba . '">
                <a href="/board?show=' . $board['id'] . '"
class="list_item_link starred" style="border-left: 40px solid ' .
$board['background'] . ';">' . $board['name'] . '
                <div class="list_item_icons icons_visible">
                    <i class="star fa fa-star"></i>
                </div>
                </a>
            </li>';
        }
    }
    echo '</ul></div>';

    $recent_boards_index = array();
    for ($i = 0; $i < count($boards); $i++) {
        $board_datetime = Carbon::createFromFormat('Y-m-d H:i:s',
$boards[$i]["last_seen"], 'Europe/Kiev');
        $now = Carbon::now('Europe/Kiev');
        $hours = $board_datetime->diffInHours($now);
        if($hours == 0) {
            array_push($recent_boards_index, $i);
        }
    }
    $has_recent_unstarred = false;
    for($i = 0; $i < count($recent_boards_index); $i++) {
        if($starred_boards_ids[$boards[$recent_boards_index[$i]]['id']]
!= 1) {
            $has_recent_unstarred = true;
        }
    }
    $recent_display_value = $has_recent_unstarred ? "block" : "none";
    echo '<div class="popup_recent_boards" style="display: ' .
$recent_display_value . ';">
        <span class="popup_list_header"><i class="popup_list_icon fa
fa-clock-o"></i> Останні дошки</span>
        <span class="toggle_board_list"><i class="minus_icon fa fa-
minus"></i></span>
        <ul class="popup_boards_list">';
    if(count($recent_boards_index) > 0) {
        for($i = 0; $i < count($recent_boards_index); $i++) {
            if($starred_boards_ids[$boards[$recent_boards_index[$i]]['id']] != 1) {
                $rgba =
hex2rgb($all_boards[$recent_boards_index[$i]]['background']);
                $rgba = "rgba(" . $rgba[0] . ", " . $rgba[1] . ", " .
$rgba[2] . ", 0.15);";
                echo '<li class="popup_boards_list_item" id="' .
$boards[$recent_boards_index[$i]]['id'] . '" style="background: ' . $rgba . '">
                    <a href="/board?show=' .
$boards[$recent_boards_index[$i]]['id'] . '" class="list_item_link"
style="border-left: 40px solid ' .
$all_boards[$recent_boards_index[$i]]['background'] . ';">' .
$all_boards[$recent_boards_index[$i]]['name'] . '
                    <div class="list_item_icons">
                        <i class="remove_board fa fa-times"></i>
                        <i class="star fa fa-star-o"></i>
                    </div>
                    </a>
                </li>';
            }
        }
    }
    echo '</ul>
</div>';

```

```

        echo '<div class="popup_my_boards">
            <span class="popup_list_header"><i class="popup_list_icon fa
fa-clone"></i> Мои доски</span>
            <span class="toggle_board_list"><i class="minus_icon fa fa-
minus"></i></span>
            <ul class="popup_boards_list">';
            foreach ($all_boards as $board) {
                $link_style = $starred_boards_ids[$board['id']] == 1 ? "starred"
: "slidable";
                $list_item_icons_style = $starred_boards_ids[$board['id']] == 1
? 'icons_visible"><i class="star fa fa-star"></i>' : "'><i class="star fa fa-
star-o"></i>';
                $rgba = hex2rgb($board['background']);
                $rgba = "rgba(" . $rgba[0] . ", " . $rgba[1] . ", " . $rgba[2] .
", 0.15)";
                echo '<li class="popup_boards_list_item" id="' . $board['id'] .
'" style="background: ' . $rgba . '">
                    <a href="/board?show=' . $board['id'] . '"
class="list_item_link ' . $link_style . '" style="border-left: 40px solid ' .
$board['background'] . ';">' . $board['name'] . '
                    <div class="list_item_icons ' . $list_item_icons_style .
'</div>
                        </a>
                    </li>';
            }
            echo '</ul>
            </div>';
            ?>
            <div class="add_board">
                <a href="#" id="popup_create_board">Создать новую доску..</a>
            </div>
        </div>
        </div>
        <script type="text/javascript" src="assets/js/libraries/jquery-
2.1.4.js"></script>
        <script type="text/javascript" src="assets/js/libraries/jquery-
ui.min.js"></script>
        <script type="text/javascript"
src="assets/js/libraries/bootstrap.min.js"></script>
        <script type="text/javascript"
src="assets/js/libraries/bootbox.min.js"></script>
        <script type="text/javascript" src="assets/js/header_line.js"></script>
        <script type="text/javascript">
            var board_id = <?php echo json_encode($_GET['show']); ?>,
                board_type = <?php echo json_encode($board_type); ?>,
                permission = <?php echo json_encode($user_permission);
?>;
        </script>
        <script type="text/javascript" src="assets/js/board.js"></script>
    </body>
</html>

```

// board.js - скрипт сторінки окремого запису

```

$(document).ready(function() {

    $(".board_star").click(function() {
        var starred = $(this).find("i").hasClass("fa-star") ? 0 : 1;
        $.ajax({
            method: "POST",
            url: "ajax/board.php",
            data: { board_change_starred: true, starred: starred, board_id: board_id }
        }).done(function(message) {
            if(message) {
                if(starred) {
                    $(".board_star").css({ 'background' : 'rgba(0, 0, 0, 0.15)' });
                    $(".board_star").find("i").removeClass("fa-star-o");
                    $(".board_star").find("i").addClass("fa-star");
                } else {
                    $(".board_star").css({ 'background' : 'none' });
                    $(".board_star").find("i").removeClass("fa-star");
                    $(".board_star").find("i").addClass("fa-star-o");
                }
            }
        });
    });

    $(".board_type a").click(function() {
        if(permission == "1") {
            board_type = board_type == "private" ? "public" : "private";
            $.ajax({
                method: "POST",
                url: "ajax/board.php",
                data: { board_change_type: true, type: board_type, board_id: board_id }
            }).done(function(message) {
                if(message) {
                    var text = board_type == "private" ? "Приватна" : "Публічна";
                    $(".board_type a").get(0).lastChild.nodeValue = ' + text;
                }
            });
        }
    });

    $(".add_new_list").focus(function() {
        $(".add_new_list_btn").css({ 'display': 'inline-block' });
        $(".add_new_list_close").css({ 'color': 'rgb(130, 129, 129)', 'display': 'inline-block' });
        $(this).removeClass("placehldr_white");
        $(this).addClass("placehldr_gray");
        $(".add_list").css({ 'background': '#e2e4e6' });
        $(".add_new_list").css({ 'color': 'rgb(130, 129, 129)', 'background': 'rgba(0,0,0,.05)', 'box-shadow': 'inset 0 1px 8px rgba(0,0,0,.15)', 'border': '1px solid #aaa' });
    });

    $(".add_new_list_close").click(function() {
        $(".add_new_list").addClass("placehldr_white");
        $(".add_new_list").removeClass("placehldr_gray");
        $(".add_list").css({ 'background': 'rgba(0, 0, 0, 0.12)' });
        $(".add_new_list").css({ 'color': '#fff', 'background': 'none', 'box-shadow': 'none', 'border': 'none' });
        $(".add_new_list_btn").css({ 'display': 'none' });
        $(".add_new_list_close").css({ 'display': 'none' });
    });

    $(".add_new_list_btn").click(function() {

```

```

var text_input = $(this).parent().find('.add_new_list'),
    list_name = text_input.val(),
    position = $(".list_wrapper").eq(-2).attr("position");
$.ajax({
    method: "POST",
    url: "ajax/board.php",
    data: { add_new_list: true, name: list_name, position: position, board_id:
board_id }
}).done(function(message) {
    if(message != false) {
        var json = JSON.parse(message),
            id = json.id,
            position = json.position;
        $(".add_list").before('\
<div class="list_wrapper col-md-3" position="' +
position + '">\
<div class="board_list" id="' + id + '">\
<div class="list_header">\
    <span>' + list_name + '</span>\
</div>\
<div class="list_content">\
</div>\
<div class="add_task">\
    <a href="#" class="show_controls">Додати
завдання..</a>\
    <input type="text" placeholder="Назва
завдання.." class="add_new_task placehldr_gray"/>\
    <input type="button"
class="add_new_task_btn" value="Додати"></input>\
    <a href="#" class="add_new_task_close"><i
class="fa fa-times"></i></a>\
        </div>\
    </div>\
</div>\
    ');
    text_input.val('');
    }
});
});

$(document).on('click', '.show_controls', function() {
    $(this).css({ 'display': 'none' });
    $(this).parent().find(".add_new_task").css({ 'display': 'block' });
    $(this).parent().find(".add_new_task_btn").css({ 'display': 'inline-
block' });
    $(this).parent().find(".add_new_task_close").css({ 'display':
'inline-block' });
    $(this).parent().find(".add_new_task").focus();
});

$(document).on('click', '.add_new_task_close', function() {
    $(this).parent().find(".show_controls").css({ 'display': 'block' });
    $(this).parent().find(".add_new_task").css({ 'display': 'none' });
    $(this).parent().find(".add_new_task_btn").css({ 'display': 'none'
});
    $(this).css({ 'display': 'none' });
});

$(document).on('click', '.add_new_task_btn', function() {
    var list_content = $(this).parent().parent().find('.list_content'),
        text_input = $(this).parent().find('.add_new_task'),
        task_name = text_input.val(),
        list_id = $(this).parent().parent().attr('id');
    console.log(task_name + " " + list_id);
    $.ajax({
        method: "POST",
        url: "ajax/board.php",
        data: { add_new_task: true, name: task_name, list_id: list_id }
    }).done(function(message) {

```

```
        if(message != false) {
            var json = JSON.parse(message);
            list_content.append('\
                <div class="list_task" id="' + json.id + '">\
                    <span>' + task_name + '</span>\
                </div>\
            ');
            text_input.val('');
        }
    });
});
```

Кафедра КБПЗ – 2021 рік

// ajax/board.php - обробник AJAX запитів сторінки окремого запису

```
<?php

$config = include "../config/db.php";
$dbh = new PDO("mysql:host=$config[host];dbname=$config[db_name]",
$config['username'], $config['password']);

if(isset($_POST['board_change_starred'])) {
    $starred = $_POST['starred'];
    $board_id = $_POST['board_id'];
    $user_id = $_COOKIE['id'];
    $sth = $dbh->prepare("UPDATE user_boards SET starred=:starred WHERE
user_id=:user_id AND board_id=:board_id");
    $sth->execute(array('starred' => $starred, 'user_id' => $user_id,
'board_id' => $board_id));
    if($sth) {
        echo true;
    } else {
        echo false;
    }
}

if(isset($_POST['board_change_type'])) {
    $type = $_POST['type'];
    $board_id = $_POST['board_id'];
    $sth = $dbh->prepare("UPDATE boards SET type=:type WHERE id=:board_id");
    $sth->execute(array('type' => $type, 'board_id' => $board_id));
    if($sth) {
        echo true;
    } else {
        echo false;
    }
}

if(isset($_POST['add_new_list'])) {
    $name = $_POST['name'];
    $board_id = $_POST['board_id'];
    $position = (int)$_POST['position'] + 1;
    $sth = $dbh->query("INSERT INTO lists VALUES (NULL, '$name', '$board_id',
'$position')");
    $lastID = $dbh->lastInsertId();
    if($sth) {
        echo json_encode(array('id' => $lastID, 'position' => $position));
    } else {
        echo false;
    }
}

if(isset($_POST['add_new_task'])) {
    $name = $_POST['name'];
    $list_id = $_POST['list_id'];
    $sth = $dbh->query("INSERT INTO tasks VALUES (NULL, '$list_id', '$name',
'', '', '', '')");
    $lastID = $dbh->lastInsertId();
    if($sth) {
        echo json_encode(array('id' => $lastID));
    } else {
        echo false;
    }
}

?>
```

```
// header_line.php - верхня панель веб-сайту
```

```
<div class="row">
  <div class="header_line" style="background-color: <?php
if($background_color) echo 'rgba(0, 0, 0, 0.1);' ?>;">
    <div class="left">
      <div class="header_boards_btn">
        <button id="boards_btn">
           Дошки
        </button>
      </div>
      <div class="header_search_bar">
        <input type="text" id="search_field" placeholder="пошук.." >
        <span class="search_icon"></span>
      </div>
    </div>
    <div class="right">
      <div class="header_notifications_btn">
        <button id="notifications_btn"></button>
      </div>
      <div class="header_user_profile">
        <span class="user_avatar" style="background-image:
url('images/user_avatars/<?php echo $photo; ?>')"></span>
        <input type="button" id="user_profile_btn" value="<?php echo $username;
?>" >
      </div>
      <div class="header_create_board">
        <button id="create_board_btn"></button>
      </div>
    </div>
    <div class="center">
      <div class="header_logo">
        <a href="/me">
          
          <span id="logo_text"> Pandya</span>
        </a>
      </div>
    </div>
  </div>
</div>
```