

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток планувальника завдань кафедри»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»
освітньо-кваліфікаційного рівня «бакалавр»

Виконавець роботи: студент групи ІТ-81 Молчанов Дмитро Андрійович

Кваліфікаційна робота бакалавра

захищена на засіданні ЕК

з оцінкою

_____ «__» _____ 2022 р.

Науковий керівник

(підпис)

к.т.н., доц., Федотова Н. А.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань

Студент _____
(підпис)

Суми-2022

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Молчанов Дмитро Андрійович

1. Тема роботи Web-додаток планувальника завдань кафедри

керівник роботи Федотова Наталія Анатоліївна, к.т.н., доцент,

затверджені наказом по університету від «27» квітня 2022 р. №0301 VI

2. Строк подання студентом роботи «11» червня 2022 р.

3. Вхідні дані до роботи технічна документація

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): аналіз предметної області, моделювання та проектування, розробка web-додатку

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

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 27.04.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підготовка специфікації	27.04.2022 – 04.05.2022	
2	Розробка шаблону web-додатку	05.05.2022 – 06.05.2022	
3	Проектування вікон додатку	09.05.2022 – 13.05.2022	
4	Розробка основних функціональних модулів	16.05.2022 – 30.05.2022	
5	Розробка бази даних	31.05.2022 – 01.06.2022	
6	Підключення додатку до бази даних	02.06.2022 – 02.06.2022	
7	Тестування	03.06.2022 – 07.06.2022	
8	Завантаження додатку на хостинг	08.06.2022 – 08.06.2022	
9	Оформлення пояснювальної записки	09.06.2022 – 13.06.2022	
10	Підготовка демонстраційного матеріалу та доповіді	14.06.2022 – 16.06.2022	
11	Захист	17.06.2022 – 17.06.2022	

Студент

(підпис)

Молчанов Д.А.

Керівник роботи

к.т.н., доц. Федотова Н.А.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток планувальника завдань кафедри».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел із 20 найменувань, трьох додатків. Загальний обсяг роботи - 104 сторінки, у тому числі 42 сторінки основного тексту, 3 сторінки списку використаних джерел, 54 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці інтерактивного web-додатку планувальника завдань для покращення взаємодії між працівниками кафедри.

У роботі проведено дослідження предметної області, огляд останніх досліджень і публікацій на тему трендів дизайну і актуальних фреймворків та проведено аналіз програмних продуктів-аналогів з подальшим порівнянням та виділенням сильних та слабких сторін кожного з додатків.

У роботі виконано структурно-функціональне моделювання, моделювання діаграми варіантів використання, проектування моделі бази даних та вибір засобів реалізації.

Результатом проведеної роботи є створений web-додаток планувальника завдань кафедри.

Практичне значення роботи полягає у оптимізації взаємодії між персоналом кафедри за рахунок використання розроблюваного додатку для таких процесів як планування робочого тижня, будь-яких офіційних та неофіційних заходів, свят тощо.

Ключові слова: ВЕБ-ДОДАТОК, КАФЕДРА, ДАНІ, HTML, CSS, JAVASCRIPT, PHP, СУБД MYSQL.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Огляд останніх досліджень і публікацій	8
1.2 Аналіз програмних продуктів-аналогів	9
1.3 Постановка задачі	14
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ	16
2.1 Структурно-функціональне моделювання	16
2.2 Проектування інформаційної моделі	19
2.3 Проектування моделі бази даних	21
3 РОЗРОБКА WEB-ДОДАТКУ	23
3.1 Архітектура програмного додатку	23
3.2 Програмна реалізація	25
3.3 Використання програмного продукту	30
3.4 Тестування веб-додатку	45
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ДОДАТОК А	51
ДОДАТОК Б	61
ДОДАТОК В	73

ВСТУП

Важко заперечити, що навколишній світ постійно пришвидшується. Те, що в минулому потребувало великої кількості часу, зараз можна виконати за коротший термін. Доставка їжі, пошта, бронювання столику в ресторані, запис до якого-небудь гуртка, виклик таксі, все це на сьогоднішні можна зробити за декілька хвилин. Зі збільшенням швидкості процесів навколо суспільства, відповідно, пришвидшується й людське світосприйняття. Розуміння та засвоєння інформації, адаптація до сучасних гаджетів, фільтрація та сортування великого обсягу різноманітної інформації досить напружують людський мозок. У наш час будь-який школяр гарно адаптується до будь-яких нововведень, батьки цих учнів оплачують покупки або комунальні рахунки не виходячи з дому, а старше покоління має можливість частіше бачитися з онуками завдяки відеозв'язку. Усе це показує як зміни навколишнього світу впливають на покоління.

Крім позитивних наслідків, у вигляді швидшої адаптації та світосприйняття, є й негативні наслідки пришвидшення процесів сучасного життя. Кожен день мозок взаємодіє з величезною кількістю інформації, яку дізнаємось від родичів, читаємо в новинах або отримуємо з відео на платформах YouTube, Facebook, Telegram та безліч інших. Усі ці дані викликають неабияке навантаження з самого ранку, а попереду ще робота або навчання, що теж потребує певної концентрації. У наслідок перенасичення інформацією, мозок забуває деяку інформацію. Інколи забуте не завдає великої шкоди, якщо забутий який-небудь цікавий факт, жарт, вірш або який зараз час, але бувають випадки, коли забувається пароль від облікового запису, мобільний номер важливого клієнта або справа, яка повинна бути виконана найближчим часом.

Для запобігання втратам важливої інформації люди створили багато додатків для часткового зменшення навантаження. Мобільні телефони

зберігають контакти друзів, родичів або колег, дані для авторизації в облікових записах – менеджер паролів, а для запису та зберігання нових справ є доречним використанням web-додатка – планувальник завдань з обов’язковою функцією контролю виконання та строків. Це дозволить зменшити навантаження на пам’ять, контролювати виконання завдань та оптимально налаштувати денний розклад.

Метою даного дослідження є контроль виконання та строків, та оптимізація таких процесів, як планування робочого тижня або проекту, організація різних заходів, за рахунок розробленого додатка планувальника завдань кафедри.

Для досягнення мети проекту необхідно виконати наступні завдання:

- визначити актуальність роботи;
- дослідити предметну область, проаналізувавши наявні аналоги та розробити основні риси унікальності майбутнього додатку;
- сформулювати технічне завдання;
- розробити дизайн оформлення майбутнього додатка;
- розробити структуру та функціонал web-додатка;
- спроектувати, створити та налаштувати базу даних для взаємодії з додатком;
- провести тестування;
- ввести в експлуатацію.

Результати роботи були представлені на конференції «ІМА – 2022» (СумДУ, 2022) [8].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

На теперішній час існує велика кількість сучасних веб-додатків які присвячені плануванню задач. Планувальник задач виконує планування і координування задач під час виконання, де задача це одиниця роботи, що виконує вказані дії. Задачі можуть виконуватися паралельно, як приклад можна привести роботу, виконувану елементами групи задач, паралельними алгоритмами і асинхронними агентами.

Для кращого розуміння предметної області, було проведено дослідження по чотирьом основним напрямкам: тренди дизайну на 2022 рік, актуальні фреймворки за 2021 рік, стисле представлення веб-додатку та список аналогів розроблюваного додатку.

Перший напрямок - тренди дизайну на 2022 рік [1]. Перелік актуальних фішок дизайну представлений у вигляді списку порад як для зовнішнього вигляду, так і для функціональної складової програми. Поради стосуються оформлення ілюстрацій, вибору тексту, палітри, пріоритетності елементів інтерфейсу та способів комунікації з клієнтом, що необхідно для створення інтуїтивно-доступного інтерфейсу додатку.

Другий напрямок відповідає за актуальні фреймворки [2]. Короткий опис та перелік основних переваг кожної з десяти комплексних бібліотек (Angular, React.js, Aurelia, Next.js, Vue.js та інші) допомагає обрати необхідну надбудову для подальшого створення веб-додатка.

Третій напрямок - для чого потрібен про веб-додаток, переваги його використання, коротка історія створення та розвиток веб-додатків викладено у статті [3]. Це ознайомлення з основами веб-програмування надає ширше представлення про призначення розроблюваного проекту та розуміння актуальності розробки у цілому.

Та четвертий напрямок з коротким описом функціональних особливостей з демонстрацією дизайну додатків представлено на сайті [4]. До перерахованих планерів належать: ToDoist, Trello, Remember The Milk, Asana, TickTick, Microsoft To-Do, Any.DO, MyLifeOrganiser, Evernote та різноманітні сервіси гугл. Сайт дає змогу проаналізувати основні тенденції у створенні планувальників та визначити необхідні функції для задоволення актуальних потреб користувачів.

1.2 Аналіз програмних продуктів-аналогів

Для аналізу аналогів було обрано 3 веб-додатки з вищевказаного сайту, а саме: ToDoist, RememberTheMilk та Google Keep.

ToDoist [5] – одна з найпопулярніших програм для контролю виконання завдань, яка більше 5 років займає вершини рейтингів серед кращих планерів. Існують мобільний і десктопний додатки та повноцінна веб-версія з неурізаним функціоналом.

Веб-додаток надає можливості:

- Додавати завдання та проекти;
- Налаштувати повторення;
- Повноцінного перегляду створеної задачі;
- Користування «швидким доступом»;
- Встановити пріоритетність завдання;
- Розподілити повноваження;
- Відстежувати процес виконання тасків;
- Графічно представити продуктивність команди;
- Архівувати завдання.

Інтерфейс планеру представлений на рис. 1.1.

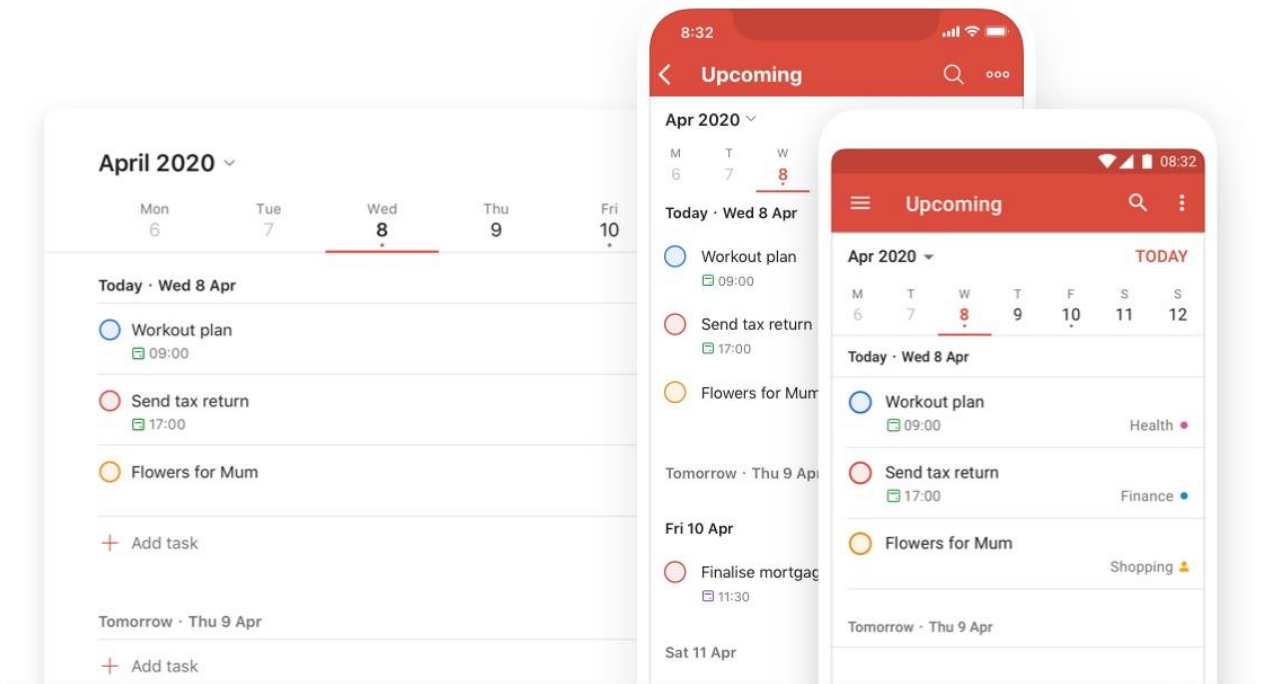


Рисунок 1.1 – Інтерфейс додатку ToDoist

Основні переваги ToDoist:

1. Інтеграція з інструментами DropBox та Google сервісами;
2. Наявність версій на більшість пристроїв: від персонального комп'ютера (ПК) до smart-годинника;
3. Офіційний сайт-лендінг наповнений усією потрібною інформацією та інструкцією користувача;
4. Показ статистики виконаних завдань та роздача «призів» при досягненні певних «міток».

Недоліки ToDoist:

1. Обмеження безкоштовної версії у вигляді 5 проєктів, 5 співробітників, 3 фільтрів завдань та 1 тижня «архіву»
2. Відсутній вбудований календар;
3. Нав'язлива реклама платних версій;
4. Проблеми з організацією підзадач та підпроєктів;
5. Технічні проблеми з перенесенням задач.

RememberTheMilk [6] – веб-планер з мінімалістичним дизайном та широкими можливостями взаємодії. Додаток дозволяє налаштувати оповіщення за допомогою Gmail, Google Calendar, Twitter, Evernote тощо. Голосові асистенти, такі як Google Assistant, Alexa та Siri, мають можливість взаємодіяти з планером. Інтерфейс веб-додатку показано на рис. 1.2.

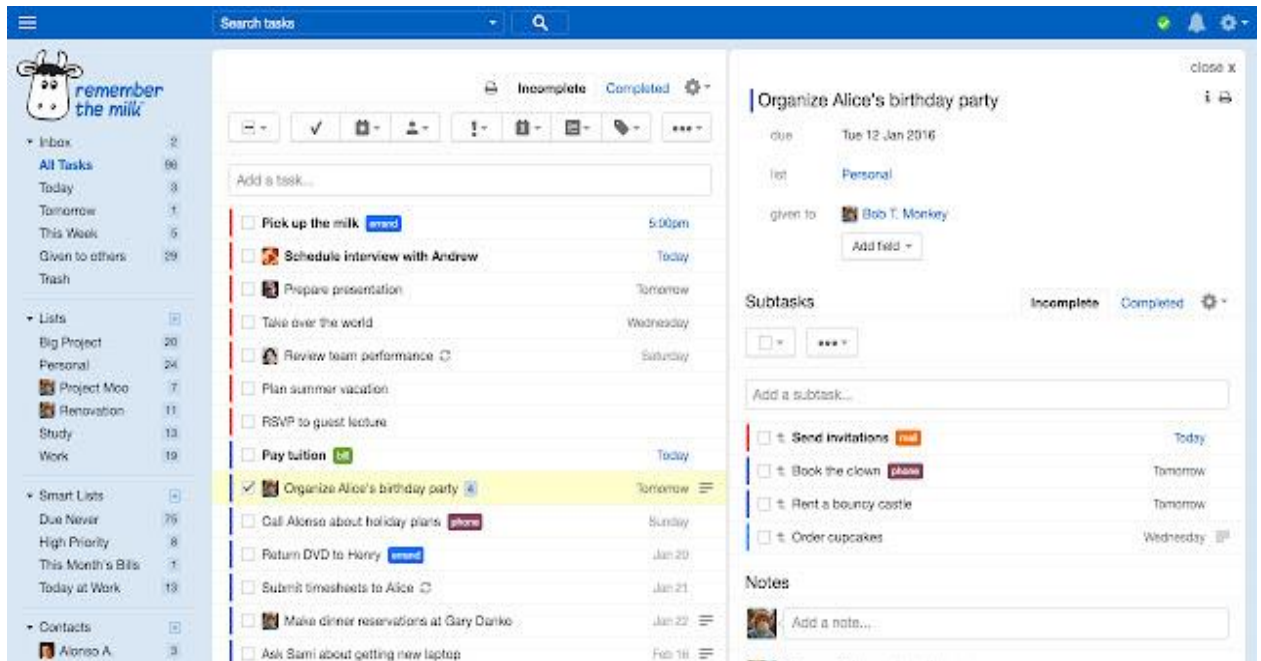


Рисунок 1.2 – Інтерфейс веб-додатку RememberTheMilk

Переваги RememberTheMilk:

1. Можливість налаштування тегів для організації різних видів завдань;
2. Широкі можливості сортування та групування;
3. Великий вибір версій програми для різних гаджетів: від ПК до smart-годинників;
4. Понад 100 тем для кастомізації інтерфейсу;
5. Можливість роботи в автономному режимі без інтернет-підключення.

Недоліки RememberTheMilk:

1. Створення підзадач, розширене сортування та групування, вибір нової теми, налаштування міток (тегів), автономний режим та багато інших істотних переваг веб-додатку доступні лише після оформлення Pro-підписки за 40 доларів у рік;
2. Погана синхронізація між мобільним додатком та веб-версією;
3. Відсутня можливість прикріплювати файли до задач;
4. Відсутній вибір мови інтерфейсу.

Google Keep [7] – повністю безкоштовний сервіс онлайн-нотаток. Користувачі мають можливість створювати статті, прописувати списки та чек-листи, а також додавати ярлики, медіафайли та ділитися списками з іншими людьми. Веб-додаток має можливість змінювати колір заміток, що дозволяє відокремлювати важливі справи від інших. Інтерфейс Google Keep представлено на рис. 1.3.

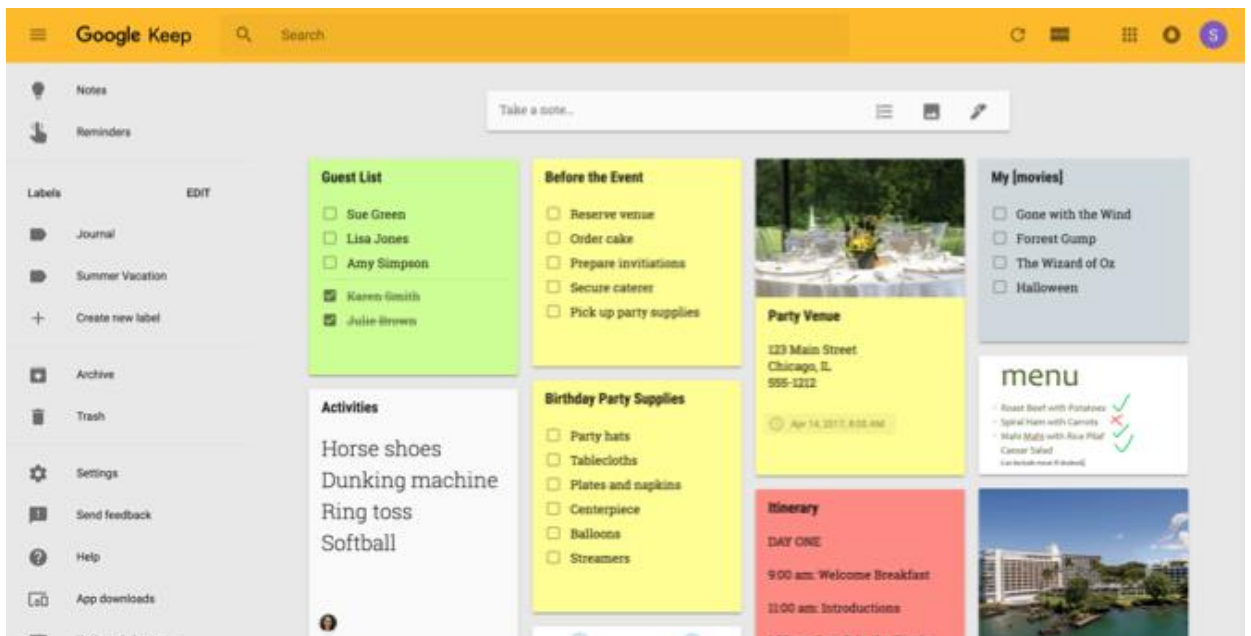


Рисунок 1.3 – Інтерфейс веб-додатку Google Keep

Переваги Google Keep:

1. Тісна інтеграція з усіма сервісами Google;

2. Доступний для усіх пристроїв;
3. Інтуїтивно зрозумілий та простий інтерфейс;
4. Наявність ярликів;
5. Наявність мобільної версії.

Недоліком Google Keep є те, що спільний доступ до розділу/нотатку можливий лише через посилання в пошті, а також порівняно невеликий функціонал.

Узагальнена порівняльна інформація про додатки представлена в таблиці 1.1.

Таблиця 1.1 – Порівняння показників існуючих аналогів

Показник/ можливість	Todoist	RememberTheMilk	Google Keep
Доступність	Безкоштовно 36\$ у рік за підписку на Pro версію 60\$ у рік за підписку на Бізнес-версію	Безкоштовно 40\$ у рік за підписку на Pro версію	Безкоштовно
Наявність десктопної версії	+	+	-
Взаємодія з сервісами Google	Google Disk	Google Disk	Усі сервіси Google

Показник/ можливість	Todoist	RememberTheMilk	Google Keep
Наявність лендінгу	+	+	-
Наявність інструкції користувача	+	+	-
Особливість додатку	Показ статистики виконаних завдань та роздача призів за досягнення	-	Візуально зрозумілий та приємний інтерфейс

1.3 Постановка задачі

Мета проекту – розробка веб-додатку планувальника задач, який зменшити навантаження на мозок користувача за рахунок планування процесів шляхом внесення завдань у планер з подальшим визначенням термінів та виконавців.

Користувачі створеного web-додатку зможуть створювати, налаштовувати та видаляти завдання, відслідковувати статистику виконання, додавати виконавців та терміни виконання завдань з будь-якого комп'ютера за рахунок зберігання резервних копій даних користувачів у хмарному сховищі.

Використання даного web-додатку дозволить оптимально планувати виконання будь-яких процесів різних напрямленостей на кафедрі: розробка проекту, організація заходів, побудова повсякденного розкладу тощо.

Розроблюваний web-додаток має бути завершено згідно встановленому графіку та доступним ресурсам. Даний продукт буде актуальним для громадян будь-якої країни за рахунок оформлення інтерфейсу українською та англійською мовами.

Назва проекту – web-додаток планувальника завдань кафедри.

Технічне завдання на розробку продукту у повному обсязі наведено у додатку А.

Для досягнення мети проекту необхідно виконати наступні завдання:

- визначити актуальність роботи;
- дослідити предметну область, проаналізувавши наявні аналоги та розробити основні риси унікальності майбутнього додатку;
- сформулювати технічне завдання
- розробити дизайн оформлення майбутнього додатка;
- розробити структуру та функціонал web-додатка;
- спроектувати, створити та налаштувати базу даних для взаємодії з додатком;
- провести тестування;
- ввести в експлуатацію.

Під час пошуку програмного забезпечення для розробки web-додатку було обрано: середовище розробки Visual Studio Code, використане для створення, редагування та зневадження вебдодатку та програмний комплекс OpenServer.

Технічне завдання наведено в додатку А.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Структурно-функціональне моделювання

IDEF0 (Function Modeling) [9] – це методологія графічного опису систем і процесів діяльності організації або будь-якої іншої системи, як безлічі взаємозалежних функцій. Функціональне моделювання дозволяє досліджувати функції організації, не пов'язуючи їх з об'єктами, що забезпечують їх реалізацію. Особливістю IDEF0 є її ієрархічне представлення об'єктів, що полегшує розуміння предметної області. В IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання в часі.

Стандарт IDEF0 представляє організацію як набір модулів, тут існує правило – найважливіша функція знаходиться у верхньому лівому кутку, крім того, існують правила сторін:

- стрілка входу завжди приходиться у ліву крайку активності;
- стрілка управління - у верхню кромку;
- стрілка механізму - нижню кромку;
- стрілка виходу - права кромка.

Функціональне моделювання web-додатку в нотації IDEF0 представлено на рисунку 2.1.



Рисунок 2.1 – Діаграма IDEF0

Перелік головних даних даної діаграми наступний:

- дані авторизації користувача: логін та пароль зареєстрованого користувача;
- запит за взаємодію з робочим полем: додавання або видалення виконавців завдання та/або додавання, редагування або видалення завдань;
- запит за ознайомлення з поточними завданнями: ознайомлення користувача з поточними завданнями, їх термінами та виконавцям.

Одним із основних понять стандарту IDEF0 є декомпозиція (Decomposition). Принцип декомпозиції застосовується при розбитті складного процесу складові його функції. У цьому рівень деталізації процесу визначається безпосередньо розробником моделі.

Декомпозиція дозволяє поступово та структуровано представляти модель системи у вигляді ієрархічної структури окремих діаграм, що робить її менш перевантаженою та легко засвоюваною.

Декомпозиція функціональної моделі можливостей адміністратора у web-додатку представлена на рисунку 2.2.

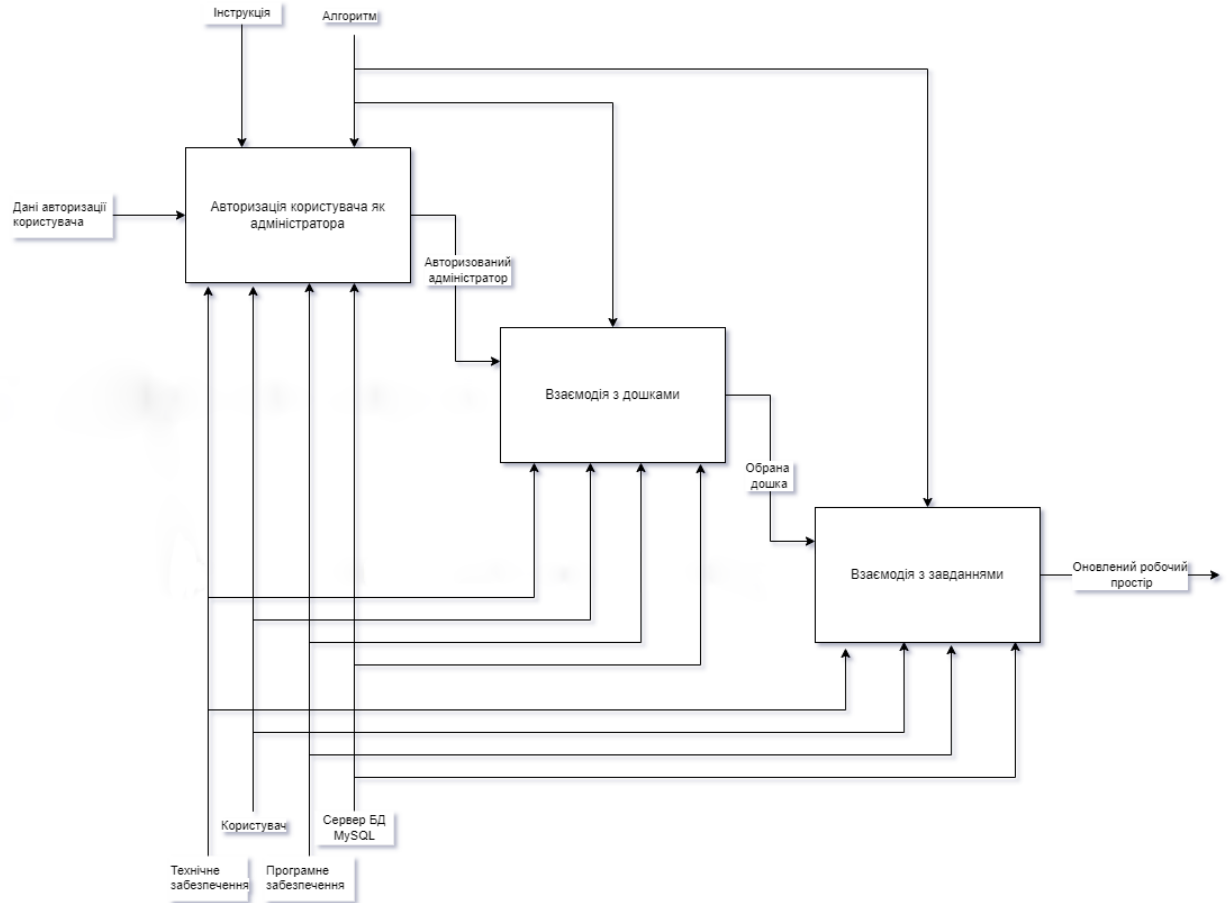


Рисунок 2.2 – Декомпозиція першого рівня

Декомпозиція функціональної моделі можливостей клієнта (персоналу) у web-додатку представлена на рисунку 2.3.

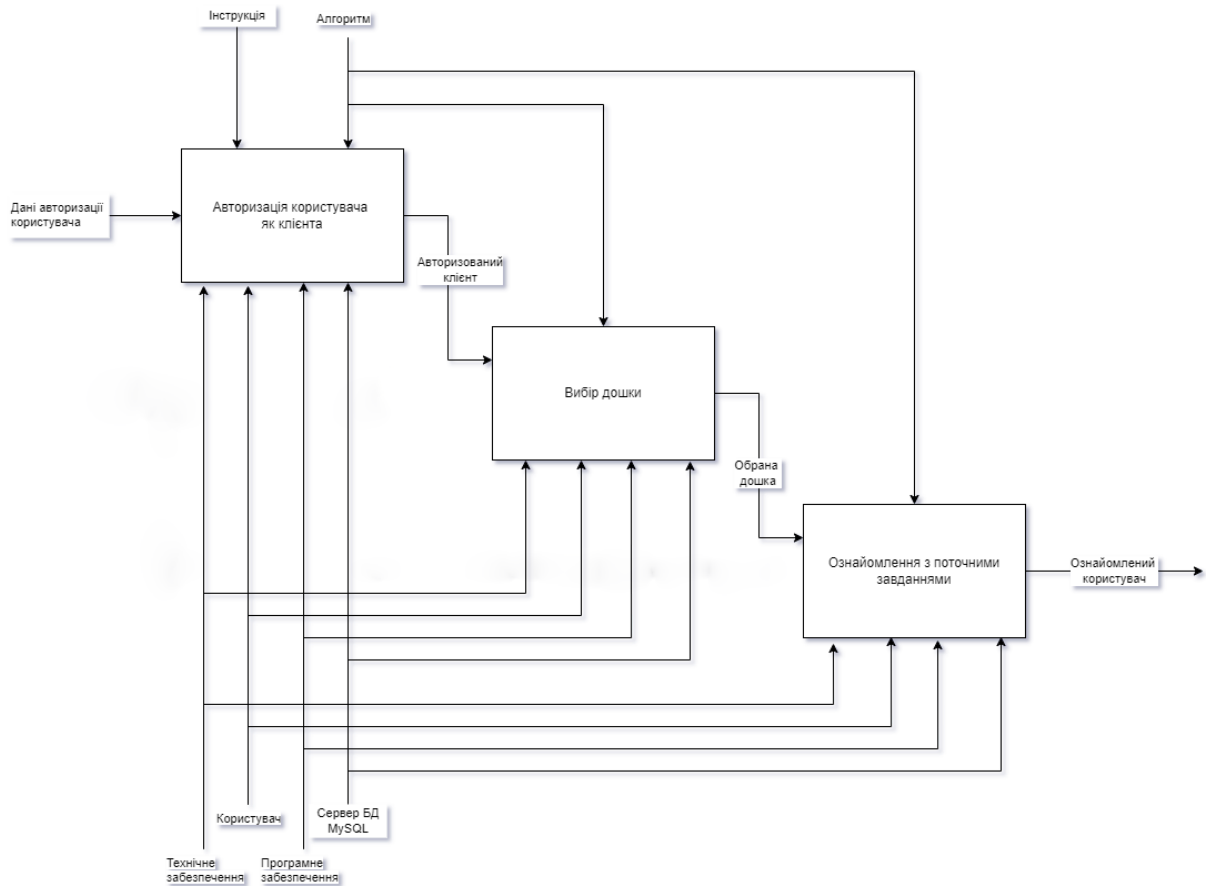


Рисунок 2.3 – Декомпозиція першого рівня

2.2 Проектування інформаційної моделі

Діаграма Прецедентів візуально зображає різноманітні сценарії взаємодії між акторами (користувачами) і прецедентами (випадками використання) [10]. Діаграма варіантів використання (ВВ) є вихідною концептуальною моделлю системи в процесі її проектування та розробки.

Добре проаналізовані і завершені специфікації перетворюються у тестові випадки.

Елементи діаграми варіантів використання:

- актор – користувач;
- прецедент – процес, тестовий випадок, варіант використання;

– граничні межі системи – позначається прямокутником на охоплює усі прецеденти системи.

Суть діаграми прецедентів полягає в тому, що розроблена система представляється як набір сутностей або акторів, які взаємодіють із системою за допомогою так званих випадків використання. Use-case використовуються для опису послуг, які система надає акторам. Іншими словами, кожен варіант використання визначає набір дій, які система виконує під час взаємодії з актором. При цьому, про те, як буде реалізована взаємодія учасників із системою, не йдеться. Діаграма варіантів використання в нотації UML представлена на рисунку 2.4.

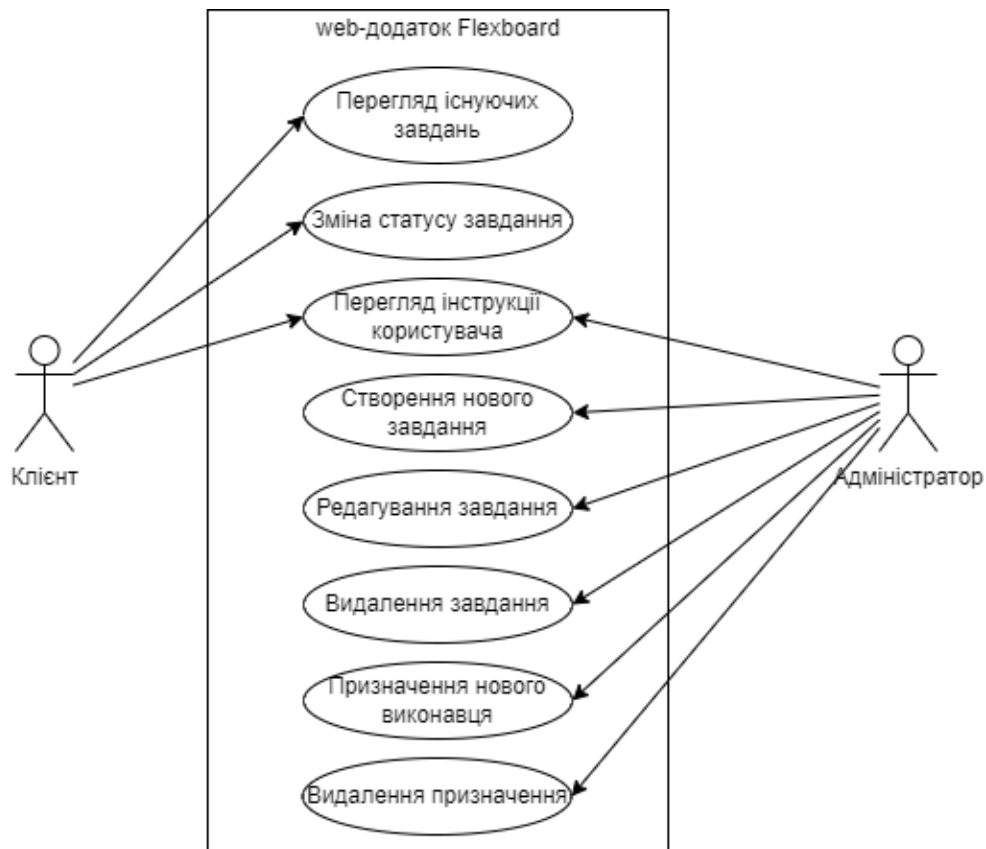


Рисунок 2.4 – Діаграма варіантів використання

Нижче приведено опис усіх елементів на рис. 2.4:

– Клієнт – користувач, який зареєстрований як персонал кафедри. Має можливість переглядати поточні завдання та взаємодіяти з ними;

– Адміністратор – користувач, який зареєстрований як завідуючий кафедрою. Має можливість переглядати, редагувати та видаляти поточні завдання, додавати та налаштовувати нові, а також призначати новостворені або існуючі завдання усім доступним виконавцям.

2.3 Проектування моделі бази даних

ER-діаграма (ERD) [11] – це різновид блок-схем, у якій показано, як різні таблиці пов'язані між собою всередині системи. ERD у більшості випадків використовується при проектуванні або налагодженні реляційних баз даних у таких сферах, як програмна інженерія, інформаційні системи для бізнесу, освіти або досліджень. ER-моделі використовують прямокутники, ромби, овали та сполучні лінії для відображення сутностей, їх атрибутів та взаємозв'язків.

В ER-діаграмах зазвичай виділяють до трьох рівнів деталізації:

– Концептуальна модель даних – схема найвищого рівня з мінімальною кількістю подробиць. Цей підхід гарний тим, що надає можливість відобразити загальну структуру моделі та архітектуру системи. Концептуальна модель не є необхідною для систем невеликого масштабу, не буде помилкою починати з логічної моделі;

– Логічна модель даних – нормалізована до четвертої нормальної форми модель, яка містить більш детальну інформацію про сутності та їх взаємозв'язки;

– Фізична модель даних – схема, створена на основі логічної моделі даних. Фізична модель повинна містити достатню кількість технічних подробиць для створення самої бази даних.

Варто пам'ятати, що ціль ER-діаграми – показати зв'язки та відносини між таблицями-сутностями, тому вони відображають тільки реляційну структуру. Також дані повинні бути чітко розбиті на поля, стовпці та рядки.

Логічну модель бази даних для розроблюваного web-додатку показано на рисунку 2.5.

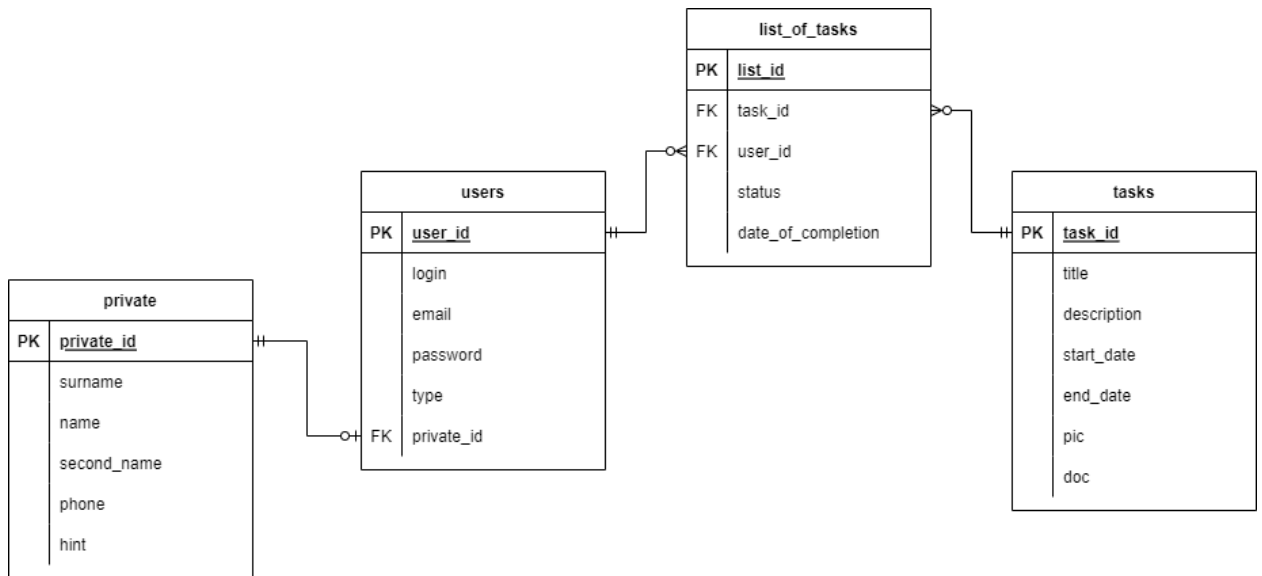


Рисунок 2.5 – Логічна модель БД для web-додатку Flexboard

В логічній моделі представлені наступні таблиці:

- **private** – таблиця, яка містить персональні дані про зареєстрованих користувачів;
- **users** – таблиця, що містить дані про зареєстровані акаунти;
- **tasks** – таблиця, у якій наведені всі створені завдання, їх опис, період виконання, а також шляхи до зображень та документів;
- **list_of_tasks** – таблиця, яка виступає зв'язним списком між **user** і **tasks** та показує статус і дату виконання дорученого завдання відповідним виконавцем.

Планування робіт та оцінка ризиків наведено в додатку Б.

3 РОЗРОБКА WEB-ДОДАТКУ

3.1 Архітектура програмного додатку

При проектуванні архітектури web-додатку було обрано технологію типу «клієнт-сервер» (CS) [12]. Основу такого типу архітектури складають два компоненти: клієнт і сервер. Ілюстрація архітектури обраного типу наведено на рис. 3.1.

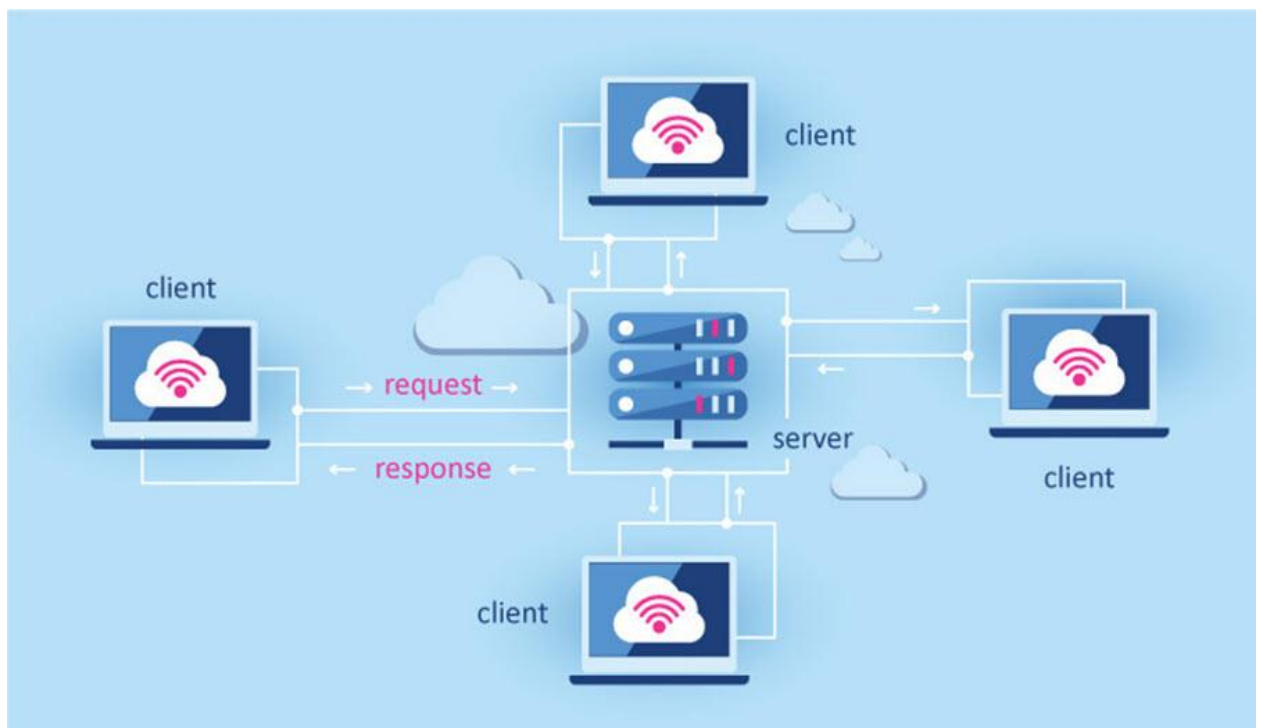


Рисунок 3.1 – Схематичне представлення клієнт-серверної архітектури

Клієнт – це комп'ютер з боку користувача, який слугує задля відправлення запиту до сервера для надання інформації або виконання певних дій.

Сервер – це комп'ютер або пристрій більш потужний ніж клієнтська машина, призначений для виконання певних завдань сервісних функцій за

бажанням клієнта, що надає користувачеві доступ до сховища певних ресурсів, інформації та баз даних.

Модель CS полягає в наступному:

1. Клієнт формує та відправляє запит на сервер;
2. Сервер отримує та оброблює запит. В разі надходження декількох, вони встановлюються в чергу і виконуються послідовно. Якщо у запитів наявні пріоритети, першими виконуються з найвищими статусами;

3. Готовий результат відправляється клієнтові.

Функції, які реалізує серверна частина:

1. Резервне копіювання та зберігання даних;
2. Надання інформації користувачам з відповідним рівнем допуску;
3. Обробка клієнтського запиту;

Функції, які реалізує клієнтська частина:

1. Формування та відправка запиту до сервера;
2. Надання користувацького інтерфейсу;
3. Отримання результату сформованого запиту, а також можливість додавати, редагувати та видаляти дані з БД за допомогою створених програмних модулів.

Архітектура розроблюваного додатку є дворівневою, яка на відміну від трирівневої використовує один сервер безпосередньо, без використання сторонніх ресурсів. Приклад дворівневої архітектури наведено на рисунку 3.2.

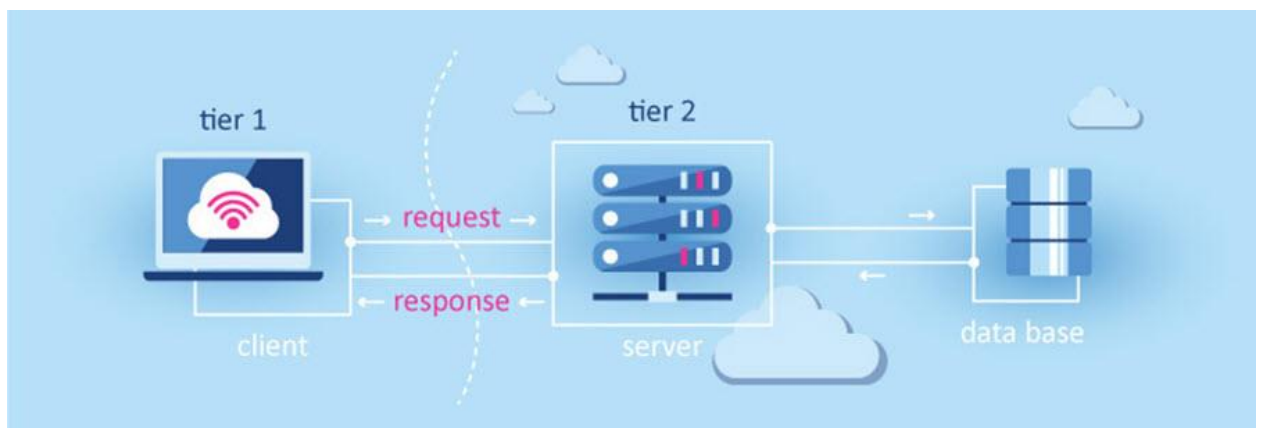


Рисунок 3.2 – Дворівнева архітектура системи

Архітектура web-додатку «Flexboard» містить наступне:

- Інтерфейс, необхідний для ознайомлення та взаємодії користувача з актуальними даними, завантаженими з БД MySQL;
- Програмний модуль для підключення інтерфейсу та інших модулів до бази даних;
- Програмні модулі для додавання, редагування та видалення даних з БД. Також містять всі необхідні перевірки, які неможливо реалізувати в формах HTML;
- База даних, яка містить у собі всю необхідну інформацію.

Структура схеми системи зображена на рисунку 3.3.

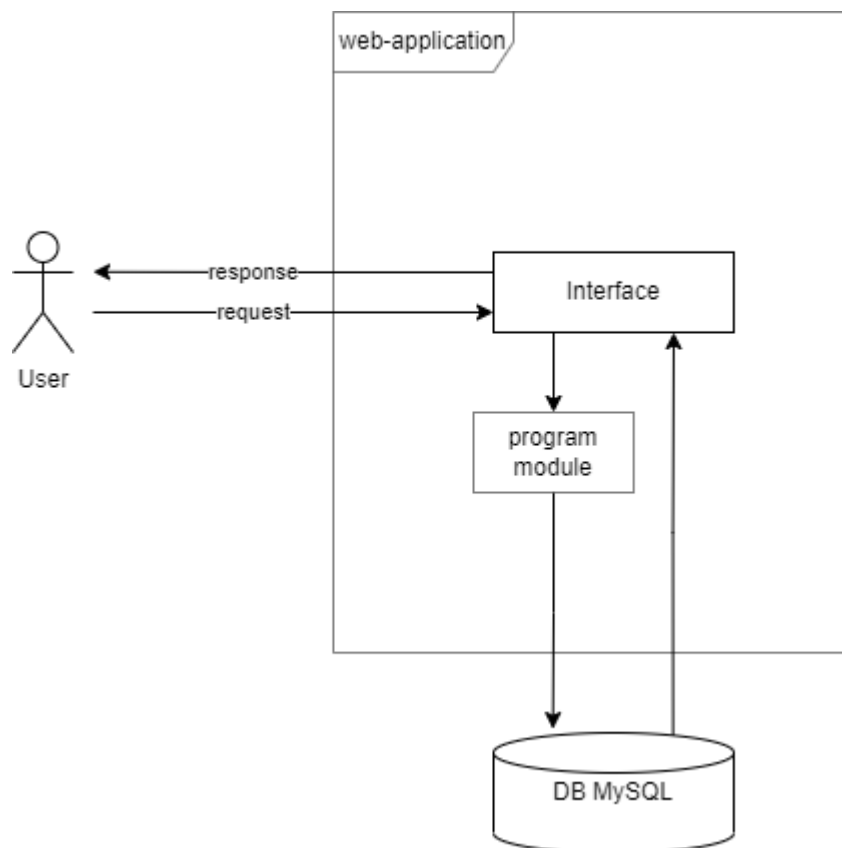


Рисунок 3.3 – Структурна схема системи

3.2 Програмна реалізація

Програмна реалізація почалася з підготовки середовища розробки. Було завантажено та встановлено програмний комплекс Open Server Panel (OSPanel), до складу якого входять локальний веб-сервер, сервер СУБД MySQL, інтерпретатор PHP та інтерфейс доступу до СУБД - PhpMyAdmin. Були створені необхідні директорії у папці D:/OpenServer/OSPanel/domains/flexboard.

Запуск OSPanel виконується з використанням панелі управління, зображеної на малюнку 3.4.

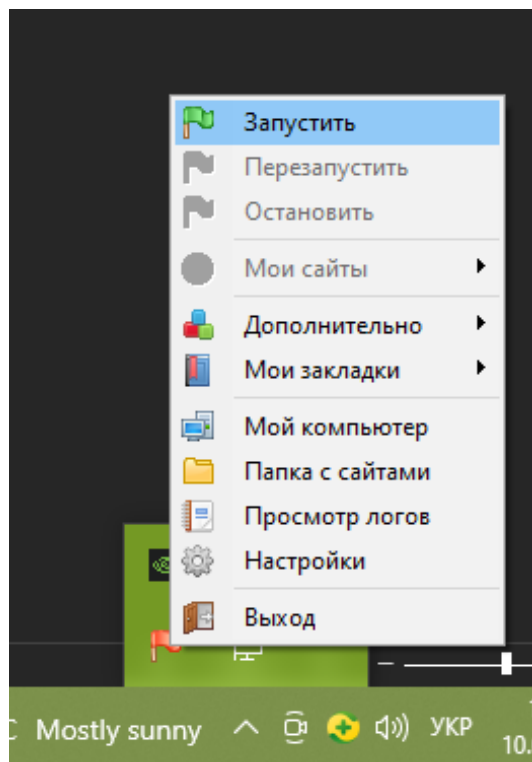


Рисунок 3.4 – Панель управління Open Server

За середовище розробки було обрано продукт компанії Microsoft – Visual Studio Code. За рахунок широкого діапазону можливостей кастомізації та дуже багатого набору інструментів, VS Code є найкращим вибором для створення нових додатків та зневаження існуючих.

Після встановлення та налаштування всього необхідного софту, було створено нову базу даних, структура якої відповідає логічній моделі, описаній у підрозділі 2.3. Список новостворених таблиць наведено на рисунку 3.5.

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> list_of_tasks	☆ [иконки]	8	InnoDB	utf8mb4_unicode_ci	48 КИБ	-
<input type="checkbox"/> private	☆ [иконки]	5	InnoDB	utf8mb4_unicode_ci	32 КИБ	-
<input type="checkbox"/> tasks	☆ [иконки]	5	InnoDB	utf8mb4_unicode_ci	16 КИБ	-
<input type="checkbox"/> users	☆ [иконки]	5	InnoDB	utf8mb4_unicode_ci	64 КИБ	-
4 таблицы	Всего	23	InnoDB	utf8mb4_unicode_ci	160 КИБ	0 Байт

Рисунок 3.5 – Список таблиць бази даних

Для зв'язку web-додатку з базою даних MySQL [14] було написано програмний модуль з використанням мови PHP. Модуль з'єднання з БД представлено на рисунку 3.6.

```

database_connect.php X index.php singup.php
d: > Сервер > OSPanel > domains > flexboard > database_connect.php
1  <?php
2
3  $servername = "localhost";
4  $database = "flexboard";
5  $username = "root";
6  $password = "root";
7
8  $connect = mysqli_connect($servername, $username, $password, $database);
9
10 if(!$connect){
11     echo "<script> alert('Connection failed: ' . mysqli_connect_error()) </script>";
12 }
13
14 }
15
16 >

```

Рисунок 3.6 – Модуль з'єднання з БД MySQL

Для реєстрації та авторизації були створені відповідні сторінки, з використанням мов HTML (мова тегів), CSS (мова стилю) та JavaScript [20]

(об'єктно-орієнтована мова програмування) (рис. 3.7 – 3.8) та онлайн сервіси для полегшення створення візуальної складової додатку [15-19]. Для перевірки, обробки та відправки даних до бази даних MySQL були створені програмні модулі з використанням скриптованої мови програмування PHP [13]. Програмні модулі зображені на рис. 3.9 – 3.10.

```

14 <script src="script.js"></script>
15 </head>
16
17 <body>
18
19 <div class="wrap">
20
21 <div class="reg_window">
22
23 <div class="header">
24 <p>
25 реєстрація
26 </p>
27 </div>
28
29 <form method="POST">
30 <p>логін</p>
31 <input form="main_form" type="text" name="login" class="login" placeholder="латиниця" required>
32
33 <p>електронна пошта</p>
34 <input form="main_form" type="email" name="email" placeholder="електронна@пошта" required>
35
36 <p>пароль</p>
37 <input form="main_form" type="password" class="pass" name="password" onchange="check()">
38 <button type="button" class="visible" id="closed" onclick="pass(0)">
39 </button>
40
41 <p>повторіть пароль</p>
42 <input form="main_form" type="password" class="pass" onchange="check()">
43 <button type="button" class="visible" id="closed" onclick="pass(1)">
44 </button>
45
46 <input form="main_form" type="reset" value="сброс">
47 <input type="button" value="наді" id="submit" style="grid-column: 2/4; font-size: 20px;" onclick="next()" disabled>
48
49 <div class="status">
50 <p id="status"></p>
51 </div>
52
53 </form>

```

Рисунок 3.7 – Web-сторінка з формами для реєстрації

```

17 </head>
18
19 <body>
20
21 <div class="wrap">
22
23 <div class="reg_window" style="height: 400px;">
24
25 <div class="header">
26 <p>
27 авторизація
28 </p>
29 </div>
30
31 <form action="singup.php" method="POST">
32 <p>логін</p>
33 <input type="text" name="login" class="login" placeholder="латиниця" required>
34
35 <p>пароль</p>
36 <input type="password" class="pass" name="password" onchange="check()">
37 <button type="button" class="visible" id="closed" onclick="pass(0)">
38 </button>
39
40 <p>повторіть пароль</p>
41 <input type="password" class="pass" onchange="check()">
42 <button type="button" class="visible" id="closed" onclick="pass(1)">
43 </button>
44
45 <input type="button" value="Реєстрація" class="reg_button" onclick="reg()">
46 <input type="submit" value="Авторизуватися" id="submit" name="submit" disabled>
47
48 <div class="status" style="margin-top: 10px">
49 <p id="status"></p>
50 </div>
51 </form>
52
53 </div>
54
55 </body>

```

Рисунок 3.8 – Web-сторінка з формами для авторизації

```

regup.php - Visual Studio Code
d:\Сервер > OSPanel > domains > flexboard > regup.php
1 <?php
2 session_start();
3
4 include "database_connect.php";
5
6 if(!empty($_POST['login'])){// чтоб POST сработал 1 раз
7
8 // Проверка логина, почты и номера телефона
9
10 $phone = $_POST['phone'];
11 $login = $_POST['login'];
12 $email = $_POST['email'];
13
14 $sql = "SELECT p.phone, u.login, u.email FROM private p, users u";// строки для проверки
15 $result = mysqli_query($connect, $sql);//отправка запроса
16 if(!$result){
17     $_SESSION['status'] = mysqli_error($connect);//если шось не сработало
18 }
19
20 while($row = mysqli_fetch_array($result)){
21
22     if($row['phone'] == $phone){
23         $_SESSION['status'] = 'Вказаний телефон вже зареєстровано';
24         echo "<script> document.location.href = 'registration.php'; </script>";
25     }
26
27     if($row['login'] == $login){
28         $_SESSION['status'] = 'Даний логін вже зареєстровано';
29         echo "<script> document.location.href = 'registration.php'; </script>";
30     }
31
32     if($row['email'] == $email){
33         $_SESSION['status'] = 'Введена електронна пошта вже зареєстрована';
34         echo "<script> document.location.href = 'registration.php'; </script>";
35     }
36 }
37
38 // Adding private information to database
39
Ln 70, Col 54  Spaces: 4  UTF-8  CRLF  PHP

```

Рисунок 3.9 – Програмний модуль створення нового акаунту regup.php

```

signup.php - Visual Studio Code
d:\Сервер > OSPanel > domains > flexboard > signup.php
1 <?php
2
3 session_start();
4
5 include "database_connect.php";
6
7 $login = $_POST['login'];
8 $password = $_POST['password'];
9 $check = false;
10 $admin = false;
11
12 $sql = "SELECT id, login, password, type FROM `users`";//данные для сравнения
13 $result = mysqli_query($connect, $sql);//отправка запроса
14 if(!$result){
15     print(mysqli_error($connect));//если шось не сработало
16 }
17
18 while($row = mysqli_fetch_array($result)){
19
20     if($row['login'] == $login){
21
22         if($row['password'] == $password){
23
24             $check = true;
25             $_SESSION['user'] = $row['id'];
26             if($row['type'] == 'admin'){
27                 $admin = true;
28             }
29             break;
30         }
31         }else{
32             $_SESSION['status'] = 'невірно введений пароль';
33             echo "<script> document.location.href = 'index.php'; </script>";
34         }
35     }
36 }
37
38
39
Ln 50, Col 7  Spaces: 4  UTF-8  CRLF  PHP

```

Рисунок 3.10 – Програмний модуль авторизації signup.php

Для кращого розуміння усіх основних можливостей web-додатку, була створена докладна інструкція, представлена на рисунку 3.11.

```

96 
97 <p>
98     Повідомлення про статус завдання буде показано в нижній частині форми.
99 </p>
100 </div>
101
102 <div class="tasks" style="display: none;">
103 
104 <p>
105     Для того, щоб відредагувати існуюче завдання, необхідно авторизуватися як завідувачий(-ча) кафедри та натиснути кнопку "редагування завдання". <br>
106     Для редагування, в базі даних повинно бути створене хоча б одне завдання.
107 </p>
108
109 
110 <p>
111     Всі створені завдання виведені в блоках у три рядки.<br>
112     Після знаходження необхідного завдання, є можливість відредагувати наведені текстові поля.
113 </p>
114
115 
116 <p>
117     Для прикріплення нового документу необхідно натиснути на кнопку вибору файлу та обрати .pdf або .doc/.docx файл.<br>
118     Розмір обраного документа не повинен перевищувати 2 мб.<br>
119     У базі даних зміниться інформація про прикріплений файл.<br>
120     Минулий документ буде видалено.
121 </p>
122
123 
124 <p>
125     Для збереження усієї відредагованої інформації та прикріпленого (за потреби) файлу у існуюче завдання, необхідно натиснути кнопку "Зберегти зміни".<br>
126     Якщо під час редагування не відбулася зміна файлу, то існуючий документ залишається прикріпленим до завдання.
127 </p>
128
129 
130 <p>
131     Після перезавантаження сторінки, відредагована інформація буде показана у відповідному блоці завдання.
132 </p>
133 </div>
134
  
```

Рисунок 3.11 – Файл instructions.html

Програмний код усіх створених модулів web-додатку наведено у додатку В.

3.3 Використання програмного продукту

При завантаженні web-додатку, клієнта зустрічає вікно авторизації, від якого залежить куди далі його буде переправлено – до панелі адміністратора чи робочого поля працівника кафедри. Вікно авторизації представлено на рисунку 3.12.

Для ознайомлення з функціоналом web-додатку створена сторінка-інструкція з докладним описом основних процесів додатку. Для переходу неї, необхідно натиснути на кнопку у правому нижньому куті. Сторінку з інструкціями для адміністратора представлено на рисунку 3.13, а для працівника кафедри – на рисунку 3.14.



Рисунок 3.12 – Вікно авторизації

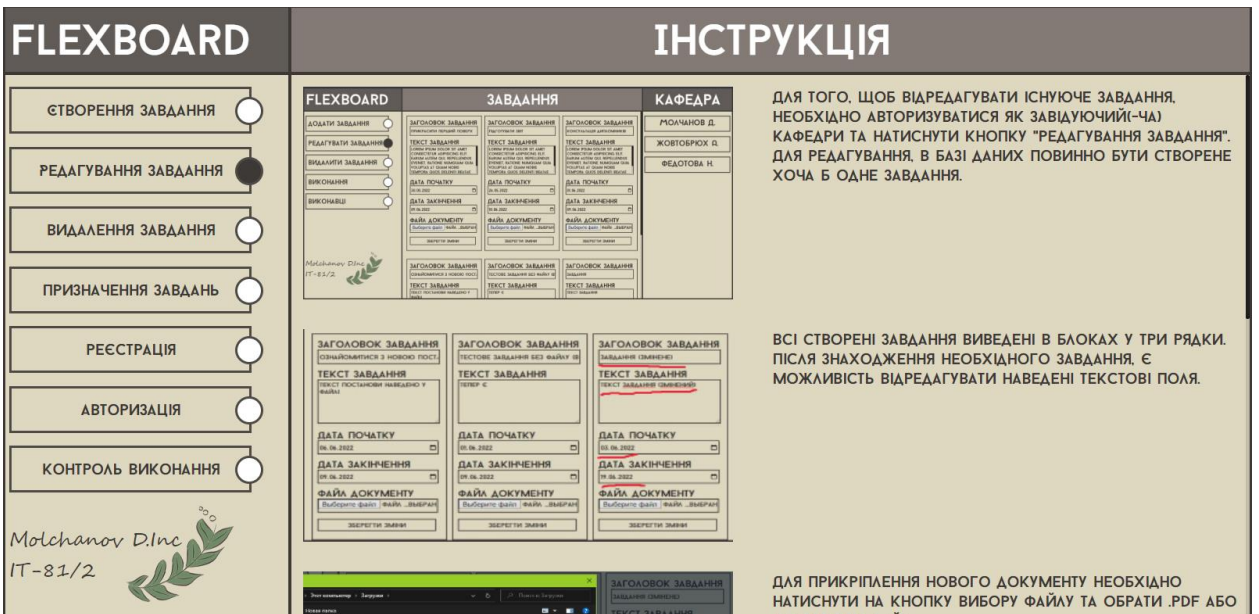


Рисунок 3.13 – Інструкція для адміністратора

FLEXBOARD	ІНСТРУКЦІЯ																									
СТВОРЕННЯ ЗАВДАННЯ	<table border="1"> <thead> <tr> <th>FLEXBOARD</th> <th>ЗАВДАННЯ</th> <th>ВИКОНАВЦІ</th> </tr> </thead> <tbody> <tr><td>ПОНЕДІЛОК</td><td></td><td>МОЛЧАНОВ Д.</td></tr> <tr><td>ВІТОРОК</td><td></td><td>ЖОВТЮРКОХ І.</td></tr> <tr><td>СЕРЕДА</td><td></td><td>ФЕДОТОВА Н.</td></tr> <tr><td>ЧЕТВЕР</td><td></td><td></td></tr> <tr><td>П'ЯТИЦЯ</td><td></td><td></td></tr> <tr><td>СУБОТА</td><td></td><td></td></tr> <tr><td>НЕДІЛЯ</td><td></td><td></td></tr> </tbody> </table>	FLEXBOARD	ЗАВДАННЯ	ВИКОНАВЦІ	ПОНЕДІЛОК		МОЛЧАНОВ Д.	ВІТОРОК		ЖОВТЮРКОХ І.	СЕРЕДА		ФЕДОТОВА Н.	ЧЕТВЕР			П'ЯТИЦЯ			СУБОТА			НЕДІЛЯ			ДЛЯ ВЗАЄМОДІЇ ЗА ПРИЗНАЧЕНИМ ЗАВДАННЯМ, НЕОБХІДНО АВТОРИЗУВАТИСЯ ЯК ПРАЦІВНИК КАФЕДРИ (USER).
FLEXBOARD	ЗАВДАННЯ	ВИКОНАВЦІ																								
ПОНЕДІЛОК		МОЛЧАНОВ Д.																								
ВІТОРОК		ЖОВТЮРКОХ І.																								
СЕРЕДА		ФЕДОТОВА Н.																								
ЧЕТВЕР																										
П'ЯТИЦЯ																										
СУБОТА																										
НЕДІЛЯ																										
РЕДАГУВАННЯ ЗАВДАННЯ	<table border="1"> <thead> <tr> <th>FLEXBOARD</th> <th>ЗАВДАННЯ</th> <th>ВИКОНАВЦІ</th> </tr> </thead> <tbody> <tr><td>ПОНЕДІЛОК</td><td>ПРИКРАСИТИ ПЕРШОЇ ПОВЕРХ 07.06.2022 ПРОГРЕС - 33% ВИКОНАВЦІ: 3</td><td>МОЛЧАНОВ Д.</td></tr> <tr><td>ВІТОРОК</td><td></td><td>ЖОВТЮРКОХ І.</td></tr> <tr><td>СЕРЕДА</td><td>КОНСУЛЬТАЦІЯ ДИТАЛОМНИКІВ 07.06.2022 ПРОГРЕС - 0% ВИКОНАВЦІ: 2</td><td>ФЕДОТОВА Н.</td></tr> <tr><td>ЧЕТВЕР</td><td>ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТАНОВОЮ 07.06.2022 ПРОГРЕС - 100% ВИКОНАВЦІ: 1</td><td></td></tr> <tr><td>П'ЯТИЦЯ</td><td></td><td></td></tr> <tr><td>СУБОТА</td><td></td><td></td></tr> <tr><td>НЕДІЛЯ</td><td></td><td></td></tr> </tbody> </table>	FLEXBOARD	ЗАВДАННЯ	ВИКОНАВЦІ	ПОНЕДІЛОК	ПРИКРАСИТИ ПЕРШОЇ ПОВЕРХ 07.06.2022 ПРОГРЕС - 33% ВИКОНАВЦІ: 3	МОЛЧАНОВ Д.	ВІТОРОК		ЖОВТЮРКОХ І.	СЕРЕДА	КОНСУЛЬТАЦІЯ ДИТАЛОМНИКІВ 07.06.2022 ПРОГРЕС - 0% ВИКОНАВЦІ: 2	ФЕДОТОВА Н.	ЧЕТВЕР	ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТАНОВОЮ 07.06.2022 ПРОГРЕС - 100% ВИКОНАВЦІ: 1		П'ЯТИЦЯ			СУБОТА			НЕДІЛЯ			ЩОБ ПОДАЧИТИ ЗАВДАННЯ, НЕОБХІДНО ОБРАТИ ПОТОЧНИЙ ДЕНЬ ТИЖНЯ НАТИСНУВШИ КНОПКУ З ВІДПОВІДНОЮ НАЗВОЮ. ПІСЛЯ ЧОГО НА ЕКРАН ВИВЕДЕТЬСЯ ВЕСЬ СПИСОК ЗАВДАНЬ, ТЕРМІН ДІЇ ЯКИХ ПРИПАДАЄ НА ОБРАНИЙ ДЕНЬ ТИЖНЯ.
FLEXBOARD	ЗАВДАННЯ	ВИКОНАВЦІ																								
ПОНЕДІЛОК	ПРИКРАСИТИ ПЕРШОЇ ПОВЕРХ 07.06.2022 ПРОГРЕС - 33% ВИКОНАВЦІ: 3	МОЛЧАНОВ Д.																								
ВІТОРОК		ЖОВТЮРКОХ І.																								
СЕРЕДА	КОНСУЛЬТАЦІЯ ДИТАЛОМНИКІВ 07.06.2022 ПРОГРЕС - 0% ВИКОНАВЦІ: 2	ФЕДОТОВА Н.																								
ЧЕТВЕР	ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТАНОВОЮ 07.06.2022 ПРОГРЕС - 100% ВИКОНАВЦІ: 1																									
П'ЯТИЦЯ																										
СУБОТА																										
НЕДІЛЯ																										
ВИДАЛЕННЯ ЗАВДАННЯ																										
ПРИЗНАЧЕННЯ ЗАВДАНЬ																										
РЕЄСТРАЦІЯ																										
АВТОРИЗАЦІЯ																										
КОНТРОЛЬ ВИКОНАННЯ																										
Molchanov D.Inc IT-81/2		ДЛЯ ОТРИМАННЯ ПОВНОЇ ІНФОРМАЦІЇ ПРО ЗАВДАННЯ, НЕОБХІДНО НАТИСНУТИ НА НЬОГО. ПІСЛЯ ЧОГО НА ЕКРАН ВИВЕДЕТЬСЯ ВЕСЬ СПИСОК ЗАВДАНЬ, ТЕРМІН ДІЇ ЯКИХ ПРИПАДАЄ НА ОБРАНИЙ ДЕНЬ ТИЖНЯ.																								

Рисунок 3.14 – Інструкція для працівника кафедри

Для повернення до вікна авторизації, необхідно натиснути на логотип «Flexboard».

Якщо користувач не зареєстрований, він може створити новий акаунт, перейшовши на сторінку з формами реєстрації. Для цього необхідно натиснути кнопку «Реєстрація». Вікно реєстрації показано на рисунку 3.15.

РЕЄСТРАЦІЯ	
ЛОГІН	ЛАТИНИЦЯ
ЕЛЕКТРОННА ПОШТА	ЕЛЕКТРОННА@ПОШТА
ПАРОЛЬ	<input type="password"/>
ПОВТОРІТЬ ПАРОЛЬ	<input type="password"/>
СБРОС	ДАЛІ

Рисунок 3.15 – Форма реєстрації нового акаунту

Після введення даних для створення нового облікового запису потрібно натиснути «Далі», щоб перейти до наступної форми реєстрації. Варто зазначити, що доки паролі не будуть введені коректно, кнопка «Далі» буде неактивною. Для того, щоб змінився статус кнопки з неактивного на активний, необхідно ввести пароль від 8 символів, вірно його повторити та клікнути за межами форми. Форма реєстрації особистих даних, представлена на рисунку 3.16.



ОСОБИСТІ ДАНІ	
ПРІЗВИЩЕ	<input type="text"/>
ІМ'Я	<input type="text"/>
ПО БАТЬКОВІ	<input type="text"/>
МОБ. ТЕЛЕФОН	<input type="text" value="000-00-00-000"/>
ПІДКАЗКА	<input type="text" value="ПІДКАЗКА ДЛЯ ПАРОЛЮ"/>
<input type="button" value="НАЗАД"/> <input type="button" value="ЗАРЕЄСТРУВАТИСЯ"/>	

Рисунок 3.16 – Форма реєстрації особистих даних

Якщо особисті дані введені коректно, відповідно до заповнювачів, користувач має натиснути кнопку «Зареєструватися», для створення нового акаунту. Після успішної реєстрації, користувача знов переносить до форми авторизації.

У випадку, якщо зареєстрований користувач має статус робітника кафедри (user), то після авторизації він переходить до робочого поля, представленого на рисунку 3.17.


FLEXBOARD	ЗАВДАННЯ	ВИКОНАВЦІ
ПОНЕДІЛОК <input type="radio"/>		МОЛЧАНОВ Д.
ВІВТОРОК <input type="radio"/>		ЖОВТОВБРЮХ Я.
СЕРЕДА <input type="radio"/>		ФЕДОТОВА Н.
ЧЕТВЕР <input type="radio"/>		
П'ЯТНИЦЯ <input type="radio"/>		
СУБОТА <input type="radio"/>		
НЕДІЛЯ <input type="radio"/>		
<i>Molchanov D.Inc</i> IT-81/2 		

Рисунок 3.17 – Робоче поле працівника кафедри

Робоче поле розділено на 3 поля:


- Дні тижня. Містить кнопки, які при натисненні виводять у друге поле завдання на обраний день (рис. 3.18);
- Завдання. Містить список завдань. При натисненні на заголовок завдання, розкриється повний зміст (рис. 3.19);
- Виконавці. Містить список зареєстрованих користувачів, яким призначене хоча б одне завдання. При натисненні на прізвище, розкривається форма з персональною інформацією (рис. 3.20).

FLEXBOARD	ЗАВДАННЯ	ВИКОНАВЦІ
ПОНЕДІЛОК <input type="radio"/>	ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ 06.06.2022 ПРОГРЕС - 33% ВИКОНАВЦІ: 3	МОЛЧАНОВ Д.
ВІВТОРОК <input checked="" type="radio"/>	КОНСУЛЬТАЦІЯ ДИПЛОМНИКІВ 06.06.2022 ПРОГРЕС - 50% ВИКОНАВЦІ: 2	ЖОВТОВБРЮХ Я.
СЕРЕДА <input type="radio"/>	ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТАНОВОЮ 06.06.2022 ПРОГРЕС - 100% ВИКОНАВЦІ: 1	ФЕДОТОВА Н.
ЧЕТВЕР <input type="radio"/>		
П'ЯТНИЦЯ <input type="radio"/>		
СУБОТА <input type="radio"/>		
НЕДІЛЯ <input type="radio"/>		
<i>Molchanov D.Inc</i> IT-81/2 		

Рисунок 3.18 – Результат натиснення кнопки дня тижня

ЗАВДАННЯ

ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ
06.06.2022 І ПРОГРЕС - 33% І ВИКОНАВЦІ: 3



LOREM IPSUM DOLOR SIT AMET CONSECTETUR
 ADIPISICING ELIT. EARUM AUTEM QUI, REPELLENDUS
 EVENIET. RATIONE NUMQUAM QUIA VOLUPTAS AT QUAM
 NOBIS TEMPORA QUOS DELENITI BEATAE ISTE VEL UNDE
 DOLORES. ENIM, TENETUR.

МОЛЧАНОВ Д.
 ЖОВТОБРЮХ А.
 ФЕДОТОВА Н.

П: 30.05.2022

К: 09.06.2022

33%

ВИКОНАНО

КОНСУЛЬТАЦІЯ ДИПЛОМНИКІВ
06.06.2022 І ПРОГРЕС - 50% І ВИКОНАВЦІ: 2

ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТАНОВОЮ
06.06.2022 І ПРОГРЕС - 100% І ВИКОНАВЦІ: 1

Рисунок 3.19 – Результат натиснення на заголовок завдання

ВИКОНАВЦІ

МОЛЧАНОВ Д.

- > МОЛЧАНОВ ДМИТРО АНДРІЙОВИЧ
- > USER@USER
- > ЗАВДАНЬ: 3

ЖОВТОБРЮХ А.

- > ЖОВТОБРЮХ АНАСТАСІЯ СЕРГІЙВНА
- > FLEX@FLEX
- > ЗАВДАНЬ: 2

Рисунок 3.20 – Результат натиснення на прізвище виконавців

Після показу повної інформації про завдання, користувач має можливість переглянути прикріплений документ, якщо такий в наявності. Індикатором документа є ілюстрація в лівій частині блоку (рис. 3.21 – 3.22).

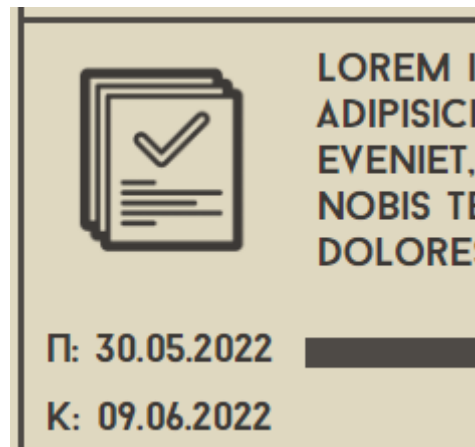


Рисунок 3.21 – Документ наявний

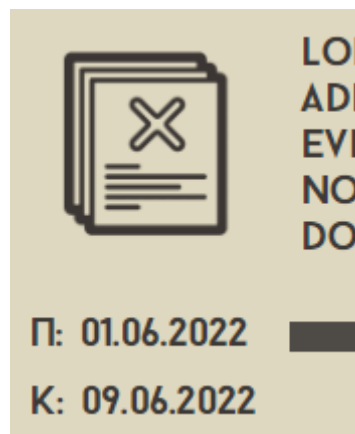


Рисунок 3.22 – Документ відсутній

Індикатором загального статусу завдання (ступінь готовності) слугує лінія прогресу (рис. 3.23), а особисто для користувача – центральна кнопка в самому низу форми. Різні стани форми представлені на рисунках 3.23 – 3.24.



Рисунок 2.23 – Лінія прогресу виконання завдання

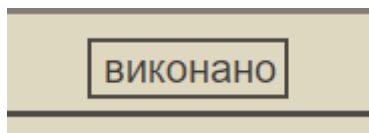


Рисунок 3.24 – Завдання не виконане

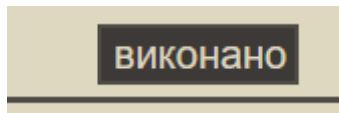


Рисунок 3.25 – Завдання виконане

Якщо користувач виконає завдання, наприклад, у вівторок, то у середу воно не відобразиться, незважаючи на термін дії. Завдання на тиждень оновлюються щонеділі.

В тому випадку, якщо користувач авторизується як адміністратор, то його перенаправить до панелі адміністратора, яка представлена на рисунку 3.26.

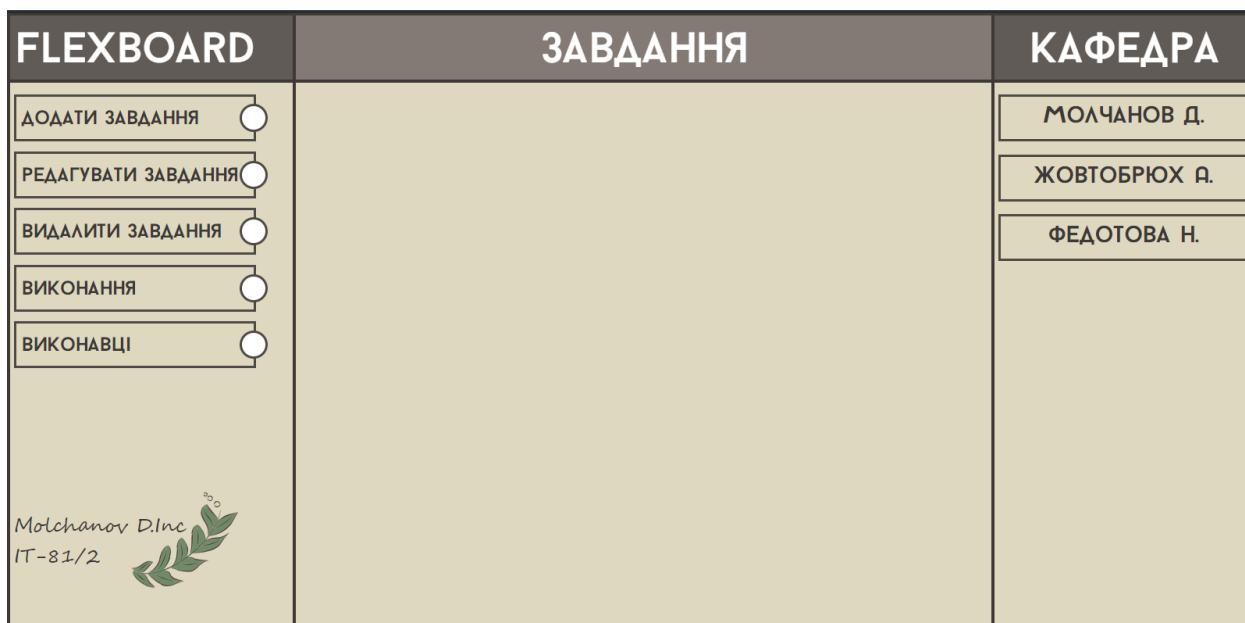


Рисунок 3.26 – Панель адміністратора

Адміністратор має можливість додавати (рис. 3.27), редагувати (рис. 3.28) та видаляти завдання (рис. 3.29), а також призначати та видаляти призначення завдань (рис. 3.30) і слідкувати за виконанням завдань

(рис. 3.31). Після введення нових даних та натиснення на будь-яку функціональну кнопку, web-сторінка оновлюється для завантаження оновленої інформації.


FLEXBOARD	ЗАВДАННЯ	КАФЕДРА
<p>ДОДАТИ ЗАВДАННЯ <input checked="" type="radio"/></p> <p>РЕДАГУВАТИ ЗАВДАННЯ <input type="radio"/></p> <p>ВИДАЛИТИ ЗАВДАННЯ <input type="radio"/></p> <p>ВИКОНАННЯ <input type="radio"/></p> <p>ВИКОНАВЦІ <input type="radio"/></p> <p>Molchanov D.Inc IT-81/2 </p>	<p>ЗАГОЛОВOK ЗАВДАННЯ <input type="text"/></p> <p>ТЕКСТ ЗАВДАННЯ <input type="text"/></p> <p>ДАТА ПОЧАТКУ <input type="text" value="ДД.ММ.ГГГГ"/></p> <p>ДАТА ЗАКІНЧЕННЯ <input type="text" value="ДД.ММ.ГГГГ"/></p> <p>ФАЙЛ ДОКУМЕНТУ <input type="text" value="Виберите файл ФАЙЛ НЕ ВИБРАН"/></p> <p><input type="button" value="СТВОРИТИ"/></p>	<p>МОЛЧАНОВ Д.</p> <p>ЖОВТОБРЮХ Д.</p> <p>ФЕДОТОВА Н.</p>

Рисунок 3.27 – Форма створення нового завдання

FLEXBOARD	ЗАВДАННЯ			КАФЕДРА
<p>ДОДАТИ ЗАВДАННЯ <input type="radio"/></p> <p>РЕДАГУВАТИ ЗАВДАННЯ <input checked="" type="radio"/></p> <p>ВИДАЛИТИ ЗАВДАННЯ <input type="radio"/></p> <p>ВИКОНАННЯ <input type="radio"/></p> <p>ВИКОНАВЦІ <input type="radio"/></p> <p>Molchanov D.Inc IT-81/2 </p>	<p>ЗАГОЛОВOK ЗАВДАННЯ ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ</p> <p>ТЕКСТ ЗАВДАННЯ LOREM IPSUM DOLOR SIT AMET CONSECTETUR ADIPISCING ELIT. EARUM AUTEM QUIL REPELLENDUS EVENIET, RATIONE NUMQUAM QUIA VOLUPTAS AT QUAM NOBIS TEMPORA QUOS DELENITI BEATAE</p> <p>ДАТА ПОЧАТКУ 30.05.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 09.06.2022</p> <p>ФАЙЛ ДОКУМЕНТУ Виберите файл ФАЙЛ _ВИБРАН</p> <p><input type="button" value="ЗБЕРЕГТИ ЗМІНИ"/></p>	<p>ЗАГОЛОВOK ЗАВДАННЯ ПІДГОТУВАТИ ЗВІТ</p> <p>ТЕКСТ ЗАВДАННЯ LOREM IPSUM DOLOR SIT AMET CONSECTETUR ADIPISCING ELIT. EARUM AUTEM QUIL REPELLENDUS EVENIET, RATIONE NUMQUAM QUIA VOLUPTAS AT QUAM NOBIS TEMPORA QUOS DELENITI BEATAE</p> <p>ДАТА ПОЧАТКУ 26.05.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 10.06.2022</p> <p>ФАЙЛ ДОКУМЕНТУ Виберите файл ФАЙЛ _ВИБРАН</p> <p><input type="button" value="ЗБЕРЕГТИ ЗМІНИ"/></p>	<p>ЗАГОЛОВOK ЗАВДАННЯ КОНСУЛЬТАЦІЯ ДИПЛОМНИКІВ</p> <p>ТЕКСТ ЗАВДАННЯ LOREM IPSUM DOLOR SIT AMET CONSECTETUR ADIPISCING ELIT. EARUM AUTEM QUIL REPELLENDUS EVENIET, RATIONE NUMQUAM QUIA VOLUPTAS AT QUAM NOBIS TEMPORA QUOS DELENITI BEATAE</p> <p>ДАТА ПОЧАТКУ 01.06.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 09.06.2022</p> <p>ФАЙЛ ДОКУМЕНТУ Виберите файл ФАЙЛ _ВИБРАН</p> <p><input type="button" value="ЗБЕРЕГТИ ЗМІНИ"/></p>	<p>МОЛЧАНОВ Д.</p> <p>ЖОВТОБРЮХ Д.</p> <p>ФЕДОТОВА Н.</p>
	<p>ЗАГОЛОВOK ЗАВДАННЯ ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТ.</p> <p>ТЕКСТ ЗАВДАННЯ ТЕКСТ ПОСТАНОВИ НАВЕДЕНО У ФАЙЛІ</p>	<p>ЗАГОЛОВOK ЗАВДАННЯ ТЕСТОВЕ ЗАВДАННЯ БЕЗ ФАЙЛУ (В)</p> <p>ТЕКСТ ЗАВДАННЯ ТЕПЕР Є</p>		

Рисунок 3.28 – Форми редагування існуючих завдань

FLEXBOARD	ЗАВДАННЯ			КАФЕДРА
<p>ДОДАТИ ЗАВДАННЯ <input type="radio"/></p> <p>РЕДАГУВАТИ ЗАВДАННЯ <input type="radio"/></p> <p>ВИДАЛИТИ ЗАВДАННЯ <input checked="" type="radio"/></p> <p>ВИКОНАННЯ <input type="radio"/></p> <p>ВИКОНАВЦІ <input type="radio"/></p>	<p>ЗАВДАННЯ</p> <p>ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ</p> <p>ДАТА ПОЧАТКУ 30.05.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 09.06.2022</p> <p>ВИДАЛИТИ ЗАВДАННЯ</p>	<p>ЗАВДАННЯ</p> <p>ПІДГОТУВАТИ ЗВІТ</p> <p>ДАТА ПОЧАТКУ 26.05.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 10.06.2022</p> <p>ВИДАЛИТИ ЗАВДАННЯ</p>	<p>ЗАВДАННЯ</p> <p>КОНСУЛЬТАЦІЯ ДИПЛОМНИКІВ</p> <p>ДАТА ПОЧАТКУ 01.06.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 09.06.2022</p> <p>ВИДАЛИТИ ЗАВДАННЯ</p>	<p>МОЛЧАНОВ Д.</p> <p>ЖОВТОБРЮХ Д.</p> <p>ФЕДОТОВА Н.</p>
<p>Molchanov D.Inc IT-81/2 </p>	<p>ЗАВДАННЯ</p> <p>ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТ.</p> <p>ДАТА ПОЧАТКУ 06.06.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 09.06.2022</p> <p>ВИДАЛИТИ ЗАВДАННЯ</p>	<p>ЗАВДАННЯ</p> <p>ТЕСТОВЕ ЗАВДАННЯ БЕЗ ФАЙЛУ (В</p> <p>ДАТА ПОЧАТКУ 01.06.2022</p> <p>ДАТА ЗАКІНЧЕННЯ 09.06.2022</p> <p>ВИДАЛИТИ ЗАВДАННЯ</p>		

Рисунок 3.29 – Форми видалення існуючих завдань


FLEXBOARD	ЗАВДАННЯ		КАФЕДРА
<p>ДОДАТИ ЗАВДАННЯ <input type="radio"/></p> <p>РЕДАГУВАТИ ЗАВДАННЯ <input type="radio"/></p> <p>ВИДАЛИТИ ЗАВДАННЯ <input type="radio"/></p> <p>ВИКОНАННЯ <input type="radio"/></p> <p>ВИКОНАВЦІ <input checked="" type="radio"/></p>	<p>ПРИЗНАЧИТИ ЗАВДАННЯ</p> <p>ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ</p> <p>ВИКОНАВЦЮ МОЛЧАНОВ ДМИТРО</p> <p>ПРИЗНАЧИТИ</p>	<p>ВИДАЛИТИ ПРИЗНАЧЕННЯ</p> <p>ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ</p> <p>ВИКОНАВЦЮ МОЛЧАНОВ ДМИТРО</p> <p>ВИДАЛИТИ</p>	<p>МОЛЧАНОВ Д.</p> <p>ЖОВТОБРЮХ Д.</p> <p>ФЕДОТОВА Н.</p>
<p>Molchanov D.Inc IT-81/2 </p>			

Рисунок 3.30 – Форми призначення завдання та видалення завдання

FLEXBOARD	ЗАВДАННЯ		КАФЕДРА
ДОДАТИ ЗАВДАННЯ <input type="radio"/>	ЗАВДАННЯ №1 ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ	ЗАВДАННЯ №2 ПІДГОТУВАТИ ЗВІТ	МОЛЧАНОВ Д.
РЕДАГУВАТИ ЗАВДАННЯ <input type="radio"/>	ВИКОНАВЦІ МОЛЧАНОВ Д. <input type="checkbox"/> - ЖОВТОВБРЮХ Д. <input checked="" type="checkbox"/> 03.06.2022 ФЕДОТОВА Н. <input type="checkbox"/> -	ВИКОНАВЦІ ФЕДОТОВА Н. <input type="checkbox"/> -	ЖОВТОВБРЮХ Д.
ВИДАЛИТИ ЗАВДАННЯ <input type="radio"/>	ЗАВДАННЯ №3 КОНСУЛЬТАЦІЯ ДИПЛОМНИКІВ	ЗАВДАННЯ №4 ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТАНОВОЮ	ФЕДОТОВА Н.
ВИКОНАННЯ <input checked="" type="radio"/>	ВИКОНАВЦІ МОЛЧАНОВ Д. <input checked="" type="checkbox"/> 07.06.2022 ЖОВТОВБРЮХ Д. <input type="checkbox"/> -	ВИКОНАВЦІ МОЛЧАНОВ Д. <input checked="" type="checkbox"/> 07.06.2022	
ВИКОНАВЦІ <input type="radio"/>	ЗАВДАННЯ №5 ТЕСТОВЕ ЗАВДАННЯ БЕЗ ФАЙЛУ (ВІДРЕДАГОВАНЕ)		
	ВИКОНАВЦІ		


Molchanov D.Inc
IT-81/2 

Рисунок 3.31 – Форма з статусами виконання завдань

Для створення нового завдання необхідно заповнити поля потрібними даними та, за потреби, прикріпити текстовий файл формату .pdf або .doc/.docx. Важно зазначити, що документ не повинен перевищувати розміром 2 МБ, в інакшому випадку виведеться повідомлення про помилку. Приклад заповненої форми представлено на рисунку 3.32. Після натиснення кнопки «Створити», новостворене завдання з'явиться у списку для редагування або видалення (рис. 3.33).

ЗАГОЛОВОК ЗАВДАННЯ	TEST
ТЕКСТ ЗАВДАННЯ	TEST
ДАТА ПОЧАТКУ	02.06.2022 <input type="checkbox"/>
ДАТА ЗАКІНЧЕННЯ	11.06.2022 <input type="checkbox"/>
ФАЙЛ ДОКУМЕНТУ	Выберите файл BACHELOR_THE...MUTENKO.DOCX
<input type="button" value="СТВОРИТИ"/>	

Рисунок 3.32 – Приклад заповнення форми нового завдання

ЗБЕРЕГТИ ЗМІНИ

ЧНЯ
ЛУ (В

ЗАГОЛОВОК ЗАВДАННЯ
TEST

ТЕКСТ ЗАВДАННЯ
TEST

ДАТА ПОЧАТКУ
03.06.2022

ДАТА ЗАКІНЧЕННЯ
11.06.2022

ФАЙЛ ДОКУМЕНТУ
Виберите файл ФАЙЛ ...ВИБРАН

ГРАН

ЗБЕРЕГТИ ЗМІНИ

Рисунок 3.33 – Створене тестове завдання

Під час редагування завдання, у користувача є можливість не тільки відредагувати текстові дані, а й змінити прикріплений файл. Для цього необхідно заново завантажити текстовий документ. Приклад зміненого завдання представлено на рисунку 3.34. Для збереження змін, необхідно натиснути кнопку «Зберегти зміни».

ЗАГОЛОВОК ЗАВДАННЯ
TEST (ЗМІНЕНИЙ)

ТЕКСТ ЗАВДАННЯ
TEST (ВІДРЕДАГОВАНИЙ)

ДАТА ПОЧАТКУ
02.06.2022

ДАТА ЗАКІНЧЕННЯ
10.06.2022

ФАЙЛ ДОКУМЕНТУ
Виберите файл 5.PDF

ЗБЕРЕГТИ ЗМІНИ

Рисунок 3.34 – Відредаговане завдання з зміненим документом

Для видалення завдання, необхідно перейти до вкладки «Видалити завдання», знайти необхідний блок та натиснути на кнопку «Видалити завдання». Вигляд форми наведено на рисунку 3.35.

ЗАВДАННЯ ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ ДАТА ПОЧАТКУ 30.05.2022 ДАТА ЗАКІНЧЕННЯ 09.06.2022 ВИДАЛИТИ ЗАВДАННЯ	ЗАВДАННЯ ПІДГОТУВАТИ ЗВІТ ДАТА ПОЧАТКУ 26.05.2022 ДАТА ЗАКІНЧЕННЯ 10.06.2022 ВИДАЛИТИ ЗАВДАННЯ	ЗАВДАННЯ КОНСУЛЬТАЦІЯ ДИПЛОМНИКІВ ДАТА ПОЧАТКУ 01.06.2022 ДАТА ЗАКІНЧЕННЯ 09.06.2022 ВИДАЛИТИ ЗАВДАННЯ
ЗАВДАННЯ ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТ. ДАТА ПОЧАТКУ 06.06.2022 ДАТА ЗАКІНЧЕННЯ 09.06.2022 ВИДАЛИТИ ЗАВДАННЯ	ЗАВДАННЯ ТЕСТОВЕ ЗАВДАННЯ БЕЗ ФАЙЛУ (В ДАТА ПОЧАТКУ 01.06.2022 ДАТА ЗАКІНЧЕННЯ 09.06.2022 ВИДАЛИТИ ЗАВДАННЯ	ЗАВДАННЯ TEST (ЗМІНЕНИЙ) ДАТА ПОЧАТКУ 02.06.2022 ДАТА ЗАКІНЧЕННЯ 10.06.2022 ВИДАЛИТИ ЗАВДАННЯ

Рисунок 3.35 – Вигляд форми видалення

У формах для призначення та видалення наявні повні списки завдань (рис. 3.36) та працівників (рис. 3.37), окрім тих, що мають статус «admin».

ПРИЗНАЧИТИ ЗАВДАННЯ ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ ПІДГОТУВАТИ ЗВІТ КОНСУЛЬТАЦІЯ ДИПЛОМНИКІВ ОЗНАЙОМИТИСЯ З НОВОЮ ПОСТАНОВОЮ ТЕСТОВЕ ЗАВДАННЯ БЕЗ ФАЙЛУ (ВІДРЕДАГОВАНЕ) TEST (ЗМІНЕНИЙ)	ВИДАЛИТИ ПРИЗНАЧЕННЯ ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ КОНАВЦЮ АЧАНОВ ДМИТРО ВИДАЛИТИ
---	--

Рисунок 3.36 – Представлений повний список завдань

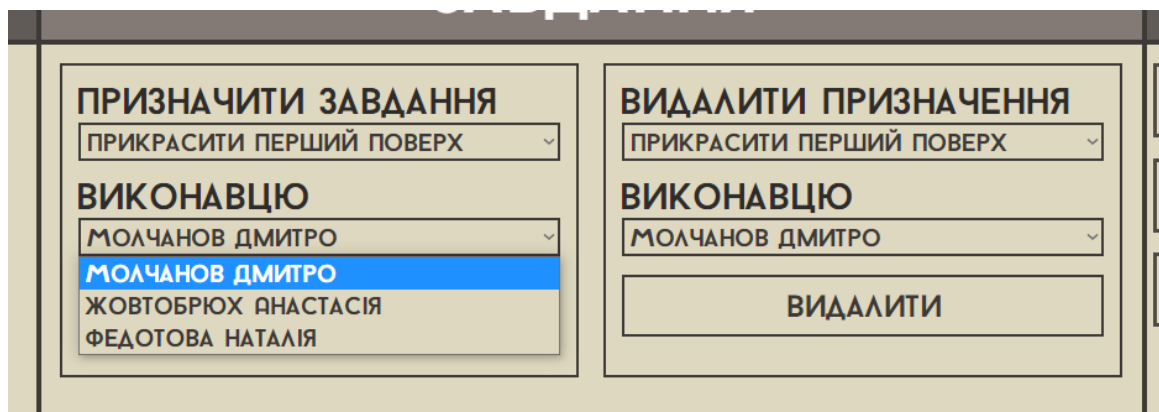


Рисунок 3.37 – Представлений повний список виконавців

Статус виконання завдання можна відслідковувати за 2 показниками: значок після прізвища виконавця (рис. 3.39 – 3.40) та дата виконання завдання. Блок виконання завдання представлено на рисунку 3.38.

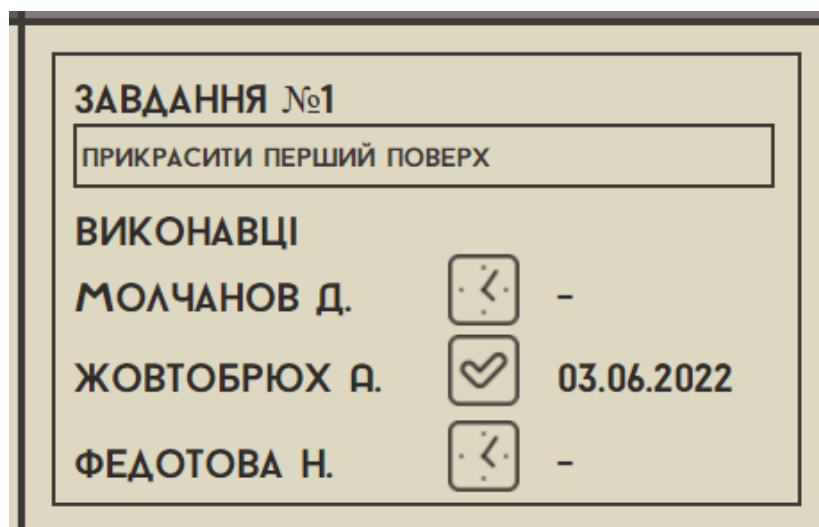


Рисунок 3.38 – Блок зі статусами виконання завдання

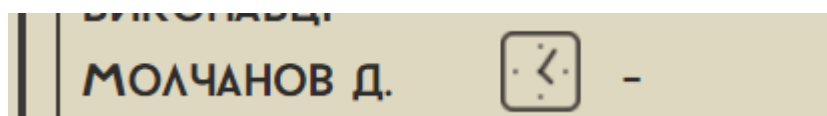


Рисунок 3.39 – Завдання не виконане

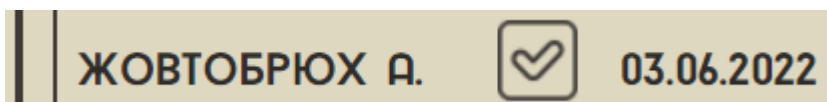


Рисунок 3.40 – Завдання виконане

3.4 Тестування web-додатку

Тестування планувальника – останній етап розробки програмного продукту. Тестування проводилося за чотирма напрямками:

- Тестування всіх посилань: посилання на зображення, логотипу web-додатку, посилання на інші модулі .php, .js та .css;
- Тестування роботи скриптів .js;
- Тестування форм на всіх сторінках: від форми реєстрації, до форми створення нового завдання;
- Тестування роботи усіх .php модулів взаємодії з базою даних MySQL.

Усі зображення відображаються коректно. Посилання на інші сторінки або текстові файли також працюють без помилок. «Рухомі» та «змінні» частини, описані в файлах .js, працюють без перебоїв. Модулі .php коректно аналізують та передають інформацію до БД MySQL. Модуль підключення до бази даних також працює справно.

Тестування форм сторінок наведено у таблиці 3.1.

Таблиця 3.1 – Тестування форм клієнтської частини системи

№	Назва тесту	Результат
1	Тест з порожніми полями у першій формі реєстрації	Перехід на наступну форму не відбувається
2	Тест з порожніми полями у другій формі реєстрації	Реєстрація не відбувається. Виводиться повідомлення 
3	Тест авторизації з невірно вказаним логіном	

Продовження таблиці 3.1

4	Тест створення завдання з порожніми полями	
5	Тест створення завдання з некоректною датою	
6	Тест створення завдання з форматом файлу, який не підтримується	

За результатами тестування можна зробити висновок, що web-додаток не має критичних помилок, а тому придатний до використання.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є розроблений web-додаток планувальника завдань кафедри.

Під час аналізу предметної області був проведений огляд останніх досліджень і публікацій, були проаналізовані існуючі методи, технології, які безпосередньо пов'язані з темою роботи. Був проведений аналіз існуючих продуктів-аналогів, які можуть бути використані для вирішення поставлених задач, з подальшим порівнянням та виділення сильних та слабких сторін кожного із додатків. В результаті вищевказаного дослідження, було сформульовано мету та задачі на розробку web-додатка планувальника завдань кафедри. Також були визначені методи та технології вирішення поставлених задач.

На етапі моделювання та проектування системи, було створено функціональну модель – діаграму IDEF0 та її декомпозицію першого рівня для кожного типу користувачів. Також було побудовано діаграму варіантів використання та логічну модель бази даних.

Під час виконання програмної реалізації web-додатку, було розроблено структурну схему системи та обрано архітектуру додатку. Також були описані основні етапи створення основних модулів web-додатку та бази даних і наведені приклади використання програмного продукту.

Розроблений web-додаток покращить взаємодію між персоналом кафедри за рахунок оптимізації процесу планування робочого тижня, офіційних та неофіційних заходів тощо. Змінам сприятимуть зручний інтерфейс та зрозумілий функціонал web-додатку.

Результати роботи були представлені на щорічній міжнародній конференції «Інформатика, математика, автоматика», ІМА-2022.

Лістинг програмного коду представлено в додатку Г.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тренди UI/UX дизайну на 2022 рік [Електронний ресурс] – Режим доступу: <https://web-promo.ua/ua/blog/trendy-uxui-dizajna-na-2022-god-15-aktualnyh-innovacij/#golovni-trendi-uiux-dizajnu-2022>
2. 10 JavaScript фреймворків для вивчення у 2021 році [Електронний ресурс] – Режим доступу: <https://tproger.ru/articles/10-javascript-frejmworkov-kotorye-stoit-vyuchit-v-2021-godu/>
3. Що точно таке веб-додаток? [Електронний ресурс] – Режим доступу: <https://uk.go-travels.com/73084-what-is-a-web-application-3486637-6454152>
4. Кращі планувальники на 2021 рік [Електронний ресурс] – Режим доступу: https://www.mojo.ua/ua/news/luchshie_planirovshchiki_10_prilozheniy_dlya_taym-menedzhmenta.html
5. ToDoist [Електронний ресурс] – Режим доступу: <https://todoist.com/ru/home>
6. RememberTheMilk [Електронний ресурс] – Режим доступу: <https://www.rememberthemilk.com>
7. Google Keep [Електронний ресурс] – Режим доступу: <https://keep.google.com/u/0/#home>
8. Молчанов Д.А., Федотова Н.А. «Web-додаток планувальника завдань кафедри»// «Інформатика, математика, автоматика»: матеріали та програма науково-технічної конференції, м. Суми, 27 – 28 квітня 2022 р. – Суми: Сумський державний університет, 2022.
URL: <https://elitconference.sumdu.edu.ua>

9. Розробка функціональної моделі // Методологія IDEF0 URL:
https://stud.com.ua/87184/ekonomika/metodologiya_idef0
(дата звернення: 24.05.2022)
10. Модель варіантів використання // Діаграми варіантів
використання URL:
https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12_2
(дата звернення: 24.05.2022).
11. What is an ER diagram? [Електронний ресурс] – Режим
доступу: <https://www.lucidchart.com/pages/er-diagrams>
12. Клієнт-серверна архітектура [Електронний ресурс] – Режим
доступу: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>
13. Php manual [Електронний ресурс] – Режим доступу:
<https://www.php.net/manual/en/index.php>
14. Посібник з MySQL [Електронний ресурс] – Режим доступу:
<https://metanit.com/sql/mysql/>
15. Color palettes generator [Електронний ресурс] – Режим доступу:
<https://colors.co>
16. Glyph online editor [Електронний ресурс] – Режим доступу:
<https://www.glyphrstudio.com/online/>
17. PNG to ICO converter [Електронний ресурс] – Режим доступу:
<https://convertio.co/ru/png-ico/>
18. Diagrams online tool [Електронний ресурс] – Режим доступу:
<https://www.diagrams.net>
19. Визначник кольору [Електронний ресурс] – Режим доступу:
<https://ru.inettools.net/image/opredelit-tsvet-piksela-na-kartinke-onlayn>

20. JavaScript Guide [Электронный ресурс] – Режим доступа:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide?retiredLocale=uk>

ДОДАТОК А
Технічне завдання

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку додатку
«Web-додаток планувальника завдань кафедри»

ПОГОДЖЕНО:

Доцент кафедри комп'ютерних наук

_____ Федотова Н.А.

Студент групи ІТ-81/2

_____ Молчанов Д.А.

Суми 2022

1 Призначення і мета створення додатку

1.1 Призначення додатку

Веб-додаток має оптимізувати процес планування робочого тижня для покращення взаємодії між завідуючим та персоналом кафедри Інформаційних Технологій (ІТП) Сумського Державного університету (СумДУ).

1.2 Мета створення

Оптимізація процесів створення, редагування, видалення та візуального представлення завдань, які виконувалися, виконуються або повинні бути виконані персоналом кафедри.

1.3 Цільова аудиторія

До цільової аудиторії можна віднести керівництво та персонал кафедри.

2 Вимоги до додатку

2.1 Вимоги до видів забезпечення

2.1.1 Вимоги для інформаційного забезпечення

Для реалізації розроблюваного веб-додатку було обрано наступні технології: мова розмітки гіпертексту HTML для створення каркасу вікон додатку та спеціальну мову стилю CSS для створення графічного оформлення сторінок. За створення основного функціоналу відповідає мова сценаріїв PHP, а за «рухому» частину інтерфейсу – динамічна об'єктно-орієнтована мова програмування JavaScript.

Для реалізації бази даних розроблюваного програмного продукту було обрано систему управління базами даних (СУБД) MySQL.

2.1.2 Вимоги до лінгвістичного забезпечення

Весь текст у веб-додатку має бути оформлений українською або англійською мовою.

2.1.3 Вимоги до програмного забезпечення

Для забезпечення стабільної роботи веб-додатку, операційна система має бути Windows 8.1 і вище. Браузер Mozilla Firefox 3.0, Google Chrome 3.0.195 або більш нові версії.

2.2 Структура додатку

2.2.1 Загальна інформація про структуру додатку

До структури web-додатку входять усі його сторінки, які є загальнодоступними, та адміністративна панель для керування завданнями та іншими налаштуваннями, яка доступна після авторизації як адміністратор.

Перелік сторінок web-додатку наступний:

- Вікно «Авторизація» містить форму для авторизації, а також кнопку-перехід до реєстрації і кнопку переходу до інструкції користувача;
- Вікно «Реєстрація» містить 2 форми для реєстрації: створення нового акаунту та реєстрація персональних даних. На першій формі міститься кнопка «Сброс» для очищення усієї вписаної інформації та кнопка «Далі» для переходу до наступної форми. На другій формі містяться кнопки «Назад» та «Зареєструватися»;
- Вікно «Інструкція» містить достатню кількість вкладок для того, щоб детально описати кожен із основних процесів у web-додатку;
- Вікно «Робоче поле» розділене на 3 вертикальні сектори. Перший сектор – «дні тижня» – містить сім кнопок на кожен день тижня. Натиснення на кожен з них виводить у другий (центральний) сектор всі завдання, призначені на цей день. У третьому секторі наведений список всіх працівників кафедри, задіяних хоча б одному завданні;
- Вікно «Панель адміністратора» також розділене на 3 сектори. Перший сектор містить кнопки, які відповідають на виклик відповідних форм,

а саме: створити завдання, редагувати завдання, видалити завдання, статистика виконання, надати/видалити виконавця.

2.2.2 Навігація

Перехід між вікнами виконується при натисканні відповідних кнопок. Вікно авторизації переводить користувача до робочого простору. В залежності від ролі користувача, стає доступною панель адміністратора для управління завданнями.

2.2.3 Дизайн та структура додатку

Дизайн додатку має бути спроектований відповідно до трендів 2021 та 2022 років. Кольорова палітра та шрифти тексту мають гармонійно поєднуватися.

Розміщення елементів робочого простору схематично показано на рисунку А.1

FLEXBOARD	ЗАВДАННЯ	ВИКОНАВЦІ
ПОНЕДІЛОК <input checked="" type="radio"/>	ПІДГОТУВАТИ ЗВІТ І 20.05.2022 <input type="checkbox"/> ПРОГРЕС - 23% І ВИКОНАВЦІ: 3	МОЛЧАНОВ Д.
ВІВТОРОК <input type="radio"/>	ПРИКРАСИТИ ПЕРШИЙ ПОВЕРХ <input type="checkbox"/> 55% І 20.05.2022 І ЗАЛИШИЛОСЬ: 3	ЖОВТЮБРИХ Н.
СЕРЕДА <input type="radio"/>		ФЕДОТОВА Н.
ЧЕТВЕР <input type="radio"/>		
П'ЯТНИЦЯ <input type="radio"/>		
СУБОТА <input type="radio"/>		
НЕДІЛЯ <input type="radio"/>		
Molchanov D.Inc 24.06.2022 IT-81/2		

Рисунок А.1 – Схема робочого поля

2.2.4 Система навігації (карта додатку)

Система навігації веб-додатку представлена на рисунку А.2.

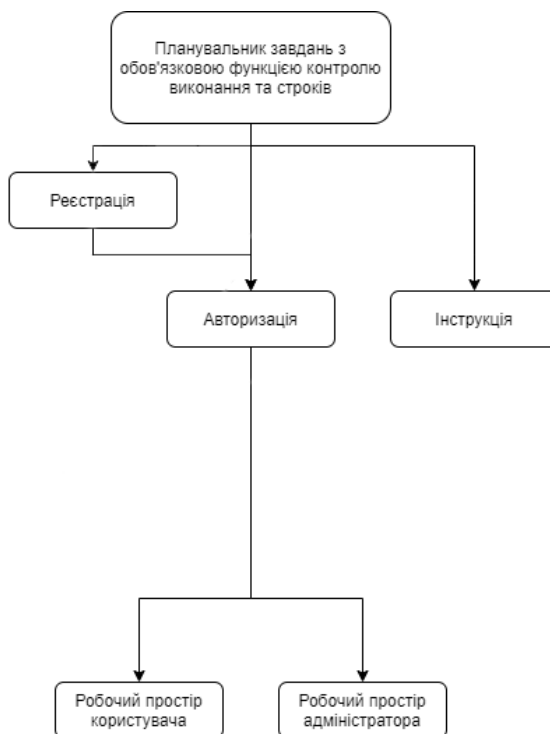


Рисунок А.2 – Карта додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Перегляд існуючих завдань	Клієнт
UN-02	Оновлення даних з БД	Клієнт
UN-03	Перегляд інструкції користувача	Клієнт
UN-04	Створення нового завдання	Адміністратор

Продовження таблиці А.1

UN-05	Видалення або додавання нового виконавця	Адміністратор
UN-06	Видалення завдання	Адміністратор
UN-07	Редагування завдання	Адміністратор
UN-08	Взаємодія з існуючим завданням	Клієнт

2.3.2 Функціональні вимоги

Проаналізувавши потреби користувачів та персоналу закладу, було визначено наступні вимоги:

- наявність реєстрації та авторизації клієнтів;
- можливість перегляду інструкції для використання функціоналу додатку;
- можливість перегляду всіх доступних завдань;
- можливість створення нового завдання з необхідними даними;
- можливість редагувати та видаляти існуючі завдання;
- можливість вільної взаємодії з виконавцями;
- можливість перегляду новоствореного робочого плану;

2.3.3 Системні вимоги

Даний розділ демонструє системні вимоги, визначені розробником. Перелік вимог наведено в таблиці А.2.

Таблиця А.2 – Системні вимоги

ІД	Системні вимоги	Пріоритет	Опис
SR-01	Наявність панелі адміністратора	М	Надає можливість адміністратору взаємодіяти з тасками

Продовження таблиці А.2

SR-02	Список завдань	М	Відображає весь список доручених завдань
SR-03	База даних планеру	М	Забезпечує синхронізацію даних планеру та гарантує актуальність інформації
SR-04	Вікно реєстрації/авторизації	М	Дозволяє створювати новий акаунт користувача або авторизуватися під існуючим
SR-05	Архів тасків	S	Відображає повну історію створених завдань
SR-06	Перемикання мови інтерфейсу	C	Дозволяє англомовним користувачам користуватися додатком

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною метою проекту.

2.4 Вимоги до додатку в цілому

2.4.1 Вимоги до структури й функціонування додатку

Веб-додаток має бути доступним через мережу Інтернет та мати 2 складові: клієнтську частину, для взаємодії користувача або адміністратора з завданнями, та базу даних, для зберігання усієї необхідної інформації про таски. Кінцевий продукт проекту має бути представлений веб-додатком, створеним з використанням мов PHP, JavaScript, CSS та HTML, який містить якісне інформаційне наповнення та сучасні графічні матеріали.

2.4.2 Вимоги до персоналу

Персонал закладу не повинен мати особливих технічних навичок для роботи з веб-додатком і його підтримкою. Єдиною вимогою є наявність навичок користування персональним комп'ютером та роботи у браузері.

2.4.3 Вимоги від зацікавлених осіб

Розроблюваний додаток повинен бути створений відповідно до календарного плану робіт. Результатом виконання проекту має бути готовий веб-додаток.

2.4.4 Вимоги до збереження інформації

Уся інформація про завдання, які представлені у веб-додатку, повинна зберігатися у базі даних, реалізованій засобами СУБД MySQL.

2.4.5 Вимоги до розмежування доступу

Розроблюваний веб-додаток має бути загальнодоступним у мережі Інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор та клієнт. Адміністратор має необмежений доступ до даних з правами перегляду, додавання, редагування та видалення завдань. Доступ до адміністративної панелі надається за спеціальним логіном та паролем. Клієнт

веб-додатку може переглядати та взаємодіяти з доступними завданнями різної направленості.

3 Склад і зміст робіт із створення додатку

Опис етапів створення веб-додатку наведений в таблиці А.3.

Таблиця А.3 – Етапи створення веб-додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Аналіз предметної області	7 днів
2	Складання технічного завдання	5 днів
3	Розробка шаблону веб-додатку	3 дні
4	Проектування вікон веб-додатку	7 днів
5	Розробка модулів додатку	18 днів
6	Створення бази даних	4 днів
7	Підключення додатку до бази даних	1 день
8	Тестування	13 днів
9	Завершення роботи	2 дні
10	Оформлення документації щодо додатку	3 дні
	Загальна тривалість робіт	63 дні

4 Вимоги до складу й змісту робіт із введення додатку в експлуатацію

Для синхронізації даних між різними користувачами, необхідно розмістити базу даних на тому ж хостингу, на якому розміщено веб-додаток. Беручи до уваги особливості хостингу, потрібно змінити запити на підключення та взаємодію з БД, для коректної роботи додатку.

ДОДАТОК Б

Планування робіт

SMART-метод деталізації мети проекту.

SMART – це метод опису мети, що включає: конкретність, вимірність, досяжність, важливість і визначеність за термінами.

SMART критерії:

- Specific – мета за системою SMART має бути ясною, враховувати специфіку, актуальну та яка має відношення до справи, бізнесу;
- Measurable – тут слід позначити число. Числове визначення, кількість в абсолютному чи процентному вигляді;
- Achievable – заявлені у підході SMART цілі мають бути одночасно і досяжні, і амбітні;
- Relevant – мета має бути узгоджена з основними цілями компанії та планами. Дії мети методом SMART повинні допомагати виконанню головної місії підприємства. Важлива також узгодженість щодо ресурсів та співробітників;
- Time bound – повинен бути встановлений кінцевий термін для проміжних і кінцевих результатів, після якого буде ясно, наскільки ми близькі до запланованого.

Результати деталізації методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Розробка десктопного додатку-планеру для оптимізації процесу планування проектів, заходів, щотижневого розкладу
-------------------------	---

Продовження таблиці Б.1

Measurable (вимірювана)	Створений план виконання або розклад з оптимально відсортованими завданнями з вказаними термінами та виконавцями
Achievable (досяжна, узгоджена)	Мета цілком досяжна, є затверджене технічне завдання
Relevant (реалістична)	Для зменшення навантаження на пам'ять користувача та запобігання негативним наслідкам від забутого завдання
Time-framed (обмежена в часі)	Є конкретний термін – до кінця 4 курсу (орієнтовно 25.05.2022)

Планування змісту структури робіт.

Ієрархічна структура робіт (Work Breakdown Structure) – інструмент, що дозволяє розбити проект на складові частини. Вона встановлює ієрархічно структуроване розподіл робіт з реалізації проекту для всіх задіяних в ньому працівників.

У ході побудови WBS здійснюється послідовна декомпозиція проекту на підпроекти, пакети робіт різного рівня, пакети детальних робіт.

Декомпозиція – це поділ результатів проекту на менші, більш керовані компоненти до рівня пакетів робіт. Пакети робіт зазвичай відповідають самому нижньому рівню деталізації і складаються з окремих робіт. Декомпозиція повинна бути коректною, тобто елементи будь-якого рівня WBS повинні бути необхідні і достатні для створення відповідного елемента верхнього рівня.

Ієрархічна структура робіт являє собою, по суті, перелік завдань проекту. Вона може бути представлена в графічному вигляді або у вигляді опису, що відображає вкладення робіт. Ієрархічна структура робіт організовує і визначає весь зміст проекту.

Роботи, не включені у WBS, не є роботами проекту. На рисунку Б.1 представлено ієрархічну структуру робіт проекту.

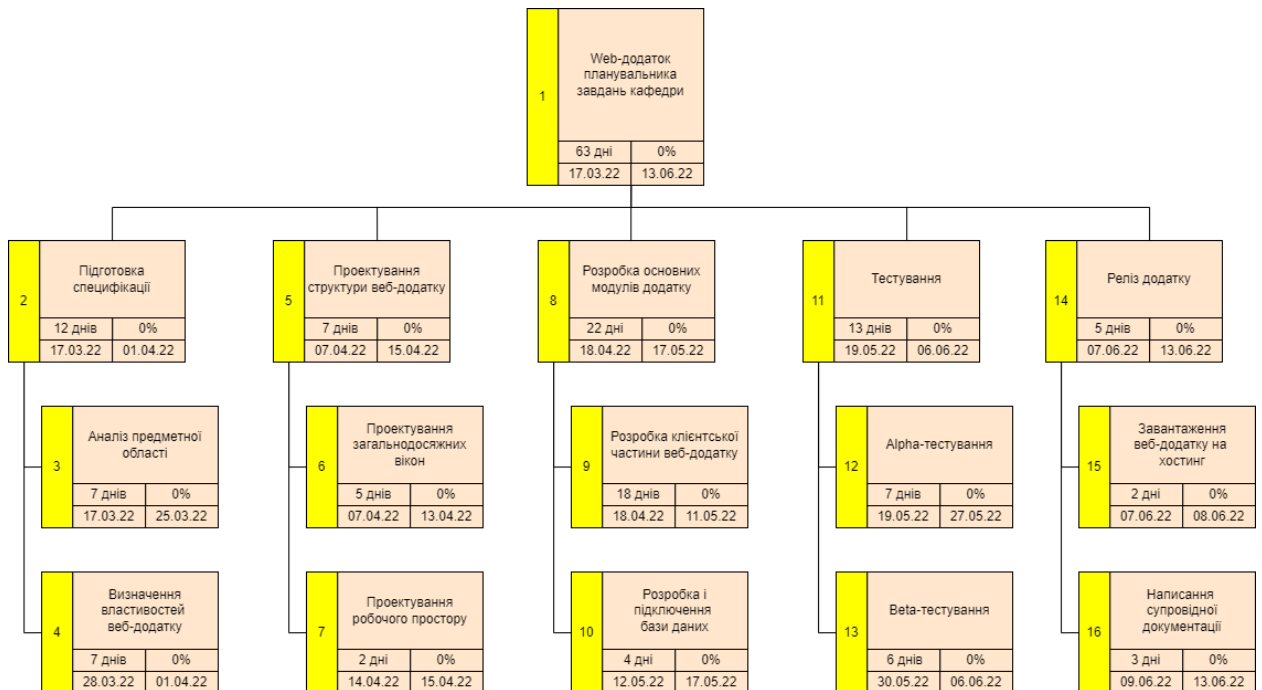


Рисунок Б.1 – WBS-структура проекту

Планування структури організації.

Наступним етапом після декомпозиції процесів є розробка організаційної структури виконавців або OBS, яка визначається як графічна структура відображення учасників або відповідальних осіб, які беруть участь у реалізації проекту.

У ролі відповідальних осіб виступають співробітники, що відповідають за організацію і виконання елементарної роботи, що зазначена у WBS. Кожну елементарну роботу можна розглядати як окремий проект.

На рисунку Б.2 представлено організаційну структуру планування проекту. Список виконавців, що функціонують в проекті описано в таблиці Б.2.

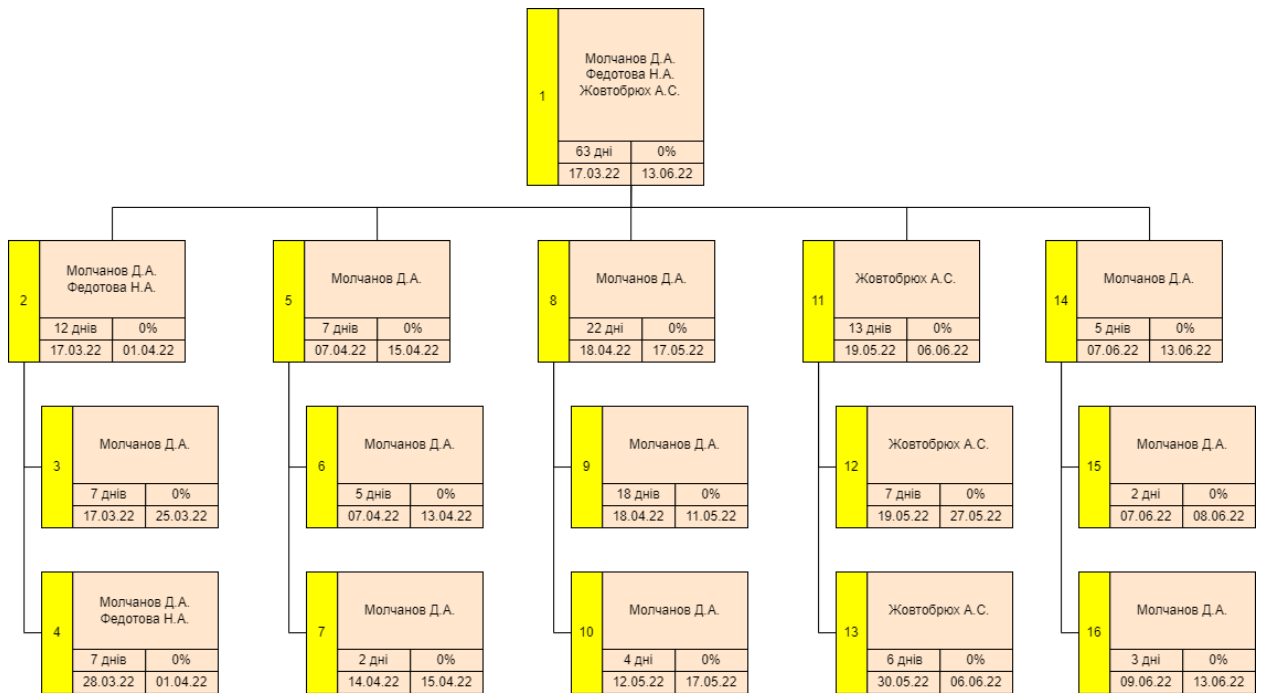


Рисунок Б.2 – OBS-структура проекту

Діаграма Ганта.

Діаграма Ганта – це популярний тип стовпчастих діаграм (гістограм), який використовується для ілюстрації плану, графіка робіт за яким-небудь проектом. Є одним з методів планування проектів. Використовується в додатках з управління проектами. На даний час Діаграма Ганта є стандартом де-факто в теорії та практиці управління проектами, принаймні, для відображення структури переліку робіт за проектом.

Діаграма Ганта являє собою відрізки, розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому проекту, задачі або підзадачі. Проекти, задачі та підзадачі, які складають план, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання.

Календарний графік проекту представлено на рисунках Б.3-Б.5.

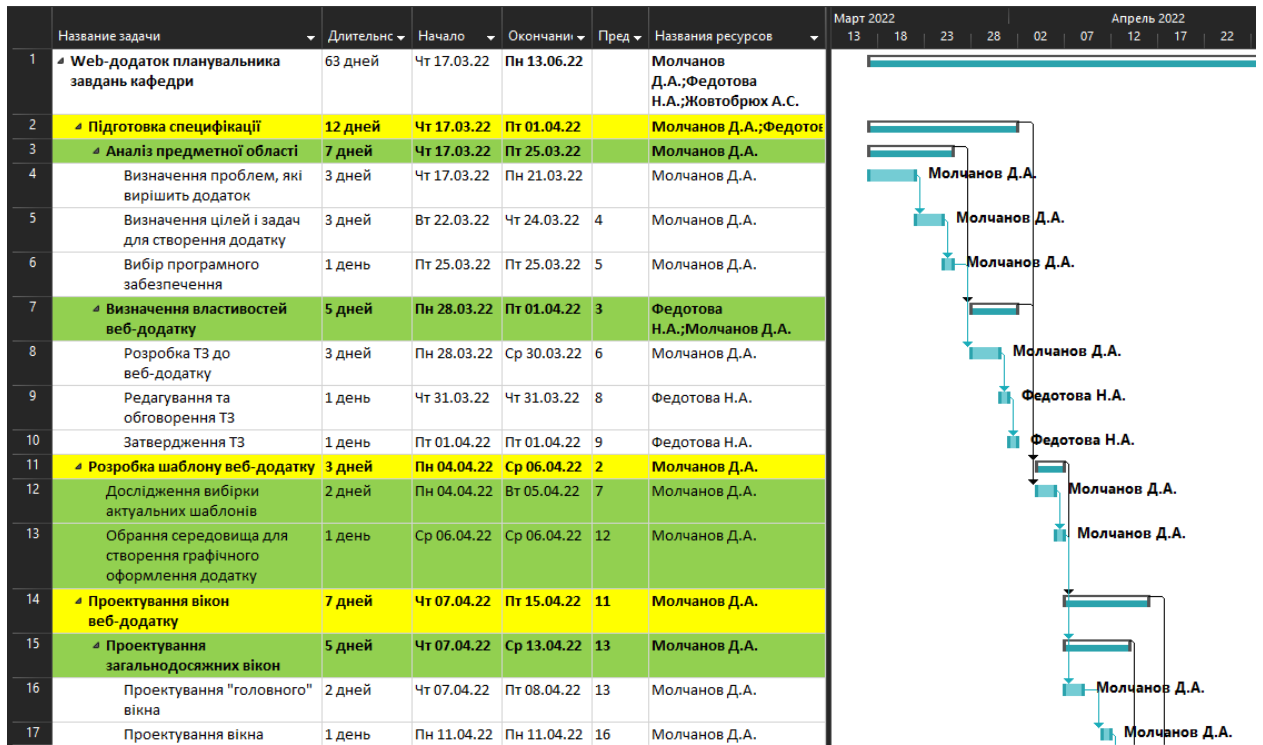


Рисунок Б.3 – Календарний графік проекту

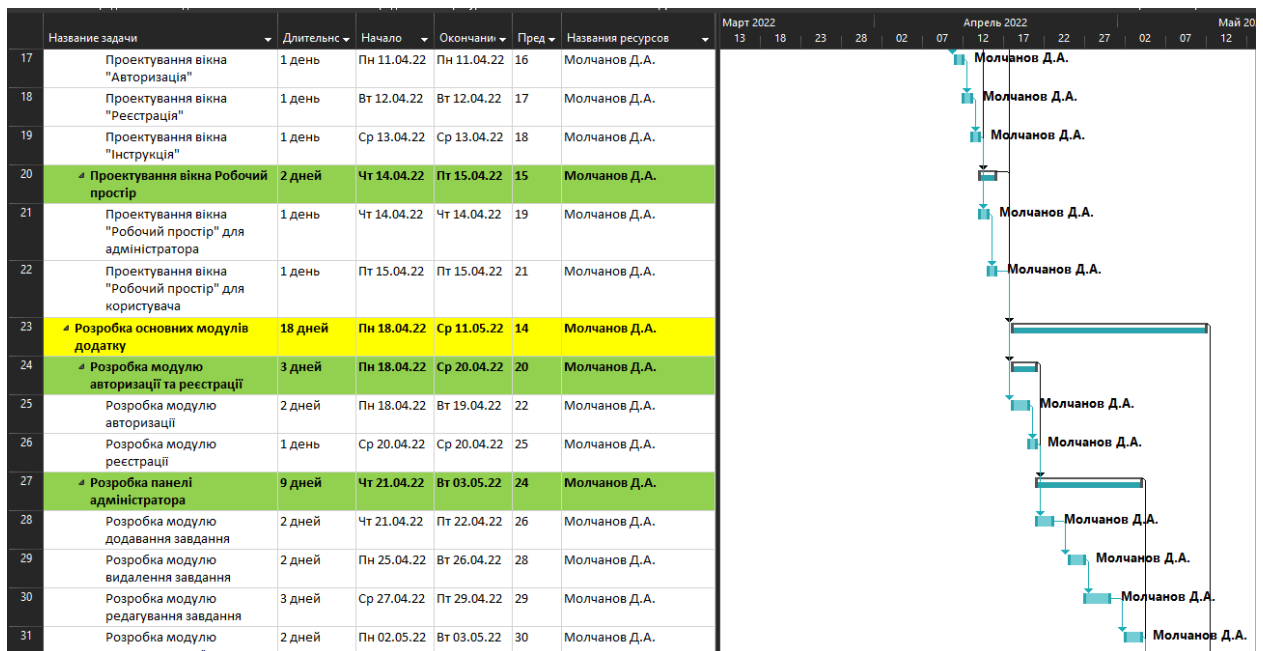


Рисунок Б.4 – Продовження календарного графіку проекту

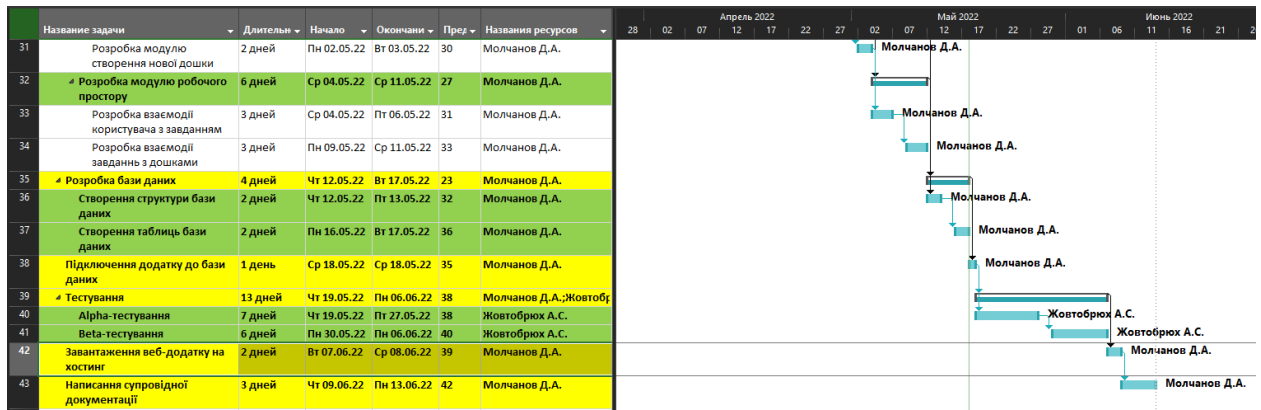


Рисунок Б.5 – Продовження графіку проекту

Управління ризиками проекту.

Управління ризиком – це процес реагування на події та зміни ризиків в процесі виконання проекту. При цьому важливим є проведення моніторингу ризиків. Моніторинг ризиків включає контроль ризиків протягом всього життєвого циклу проекту. Якісний моніторинг ризиків забезпечує управління інформацією, яка допомагає приймати ефективні рішення до настання ризикових подій. При розробці будь-якого проекту необхідно передбачити всі можливі ризики, які можуть трапитись та провести необхідні заходи для їх уникнення чи зменшенню збитків.

У таблиці Б.3 представлено шкалу для класифікації ризиків за величиною впливу на проект та ймовірністю виникнення.

Таблиця Б.3 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу.

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Оцінювання виконується за показниками, що описані в таблиці Б.3. Визначені ризики для проекту наведено в таблиці Б.4. У результаті планування реагування було

отримано матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рисунку Б.6. Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі. Матриця ризиків наведена у таблиці Б.5.

Таблиця Б.4 – Оцінка ймовірності виникнення, величини витрат та індексу ризику.

№	Назва ризику	Ймовірність (0,1 – 0,9)	Вплив (0,05-0,8)	Ранг
1	Низька кваліфікація розробника	0,3	0,4	0,12
2	Неоптимальний розподіл часу	0,5	0,4	0,2
3	Часте внесення змін у ТЗ	0,3	0,2	0,06
4	Нещасні випадки	0,1	0,2	0,02
5	Поява альтернативного продукту	0,3	0,2	0,06
6	Непорозуміння між розробником та куратором	0,1	0,2	0,02
7	Нечітке завдання на розробку	0,5	0,4	0,2
8	Втрата даних	0,3	0,8	0,24

Продовження таблиці Б.4

9	Поломка обладнання	0,1	0,8	0,08
10	Збій у роботі програмного забезпечення для розробки	0,3	0,4	0,12

Таблиця Б.5 – Матриця ризиків

Ймовірність ризику (Й)	Вплив загрози (ризик)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9					
0,7					
0,5				R2(0.2), R7(0.2)	
0,3			R3(0.06), R5(0.06)	R1(0.12), R10(0.12)	R8(0.24)
0,1			R4(0.02), R6(0.02)		R9(0.08)

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці Б.6. У таблиці Б.7 описано ризики та стратегії реагування на кожен з них.

Таблиця Б.6 – Шкала оцінювання за рівнем ризику.

№	Назва	Межі	Ризики
1	Прийнятні	$0,005 \leq R \leq 0,05$	4, 6
2	Виправдані	$0,05 < R \leq 0,14$	1, 3, 5, 9, 10
3	Недопустимі	$0,14 < R \leq 0,72$	2, 7, 8

На основі даних з матриці, було створено стратегії реагування на можливі ризики. Стратегії описані в таблиці Б.7.

Таблиця Б.7. – Ризики та стратегії реагування.

ID	Опис ризику	План А	Тип стратегії реагування	План Б
RS_1	Низька кваліфікація розробника	<ol style="list-style-type: none"> 1. Забезпечити необхідну теоретичну базу для вирішення можливих питань 2. Налагодити зв'язок для консультації з куратором 	Зменшення	Проконсультуватися з куратором на предмет вирішення питання
RS_2	Неоптимальний розподіл часу	<ol style="list-style-type: none"> 1. Створити та затвердити з куратором оптимальний графік. 2. Дотримуватися затвердженого графіку 	Попередження	Внести зміни до календарного плану, проінформувати куратора та замовника.

Продовження таблиці Б.7.

RS_3	Часте внесення змін у ТЗ	1. Затвердити основний напрям додатку	Зменшення	Затвердити внесення основних змін та поправок.
RS_4	Нещасні випадки (НВ)	1. Забезпечити можливість перенести роботи на деякий період 2. Знайти «страхуючий персонал», який допоможе в розробці у разі виникнення НВ	Зменшення	Залучити додаткового розробника на час непрацездатності основного розробника.
RS_5	Поява альтернативного продукту	1. Провести попереднє дослідження альтернативних продуктів 2. Вибрати унікальну стратегію розробки	Ухилення	-

Продовження таблиці Б.7.

RS_6	Непорозуміння між розробником та куратором	1. Налагодити відносини між розробником та керівником 2. Створити комфортні умови для співпраці	Зменшення	При виявленні непорозуміння потрібно в'яснити, що саме стало причиною непорозуміння обговорити її
RS_7	Нечітке завдання на розробку	1. За вимогою, скласти глосарій для запобігання розбіжностей у розумінні слів та термінів	Передача	При виявленні невідповідності, потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки.
RS_8	Втрата даних	1. Створення резервних копій 2. Активне використання хмарного сховища	Зменшення	Завантажити останню збережену версію та продовжити розробку.
RS_9	Поломка обладнання	1. Забезпечити персонал запасним обладнанням	Ухилення	-

Продовження таблиці Б.7.

RS_10	Збій у роботі програмного забезпечення для розробки	1. Залучити спеціаліста для усунення збоїв	Зменшення	Обрати інше програмне забезпечення
-------	---	--	-----------	------------------------------------

ДОДАТОК В

ЛІСТИНГ ПРОГРАМНОГО КОДУ ОСНОВНИХ МОДУЛІВ WEB-ДОДАТКУ

Index.php

```

<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Authorization</title>
  <link rel="shortcut icon" href="regPic/icon.ico" type="image/x-icon">
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
</head>
<body>
  <div class="wrap">
    <div class="reg_window" style="height: 400px;">
      <div class="header">
        <p>
          авторизація
        </p>
      </div>
      <form action="singup.php" method="POST">
        <p>логін</p>
        <input type="text" name="login" class="login" placeholder="латиниця" required>
        <p>пароль</p>
        <input type="password" class="pass" name="password" onchange="check()">
        <button type="button" class="visible" id="closed" onclick="pass(0)">
        </button>
        <p>повторіть пароль</p>
        <input type="password" class="pass" onchange="check()">
        <button type="button" class="visible" id="closed" onclick="pass(1)">
        </button>
        <input type="button" value="Регстрація" class="reg_button" onclick="reg()">
        <input type="submit" value="Авторизуватися" id="submit" name="submit" disabled>
      </form>
    </div>
  </div>

```

```

        <div class="status" style="margin-top: 10px">
            <p id="status"></p>
        </div>
    </form>
</div>
<div class="instruction">
    <a href="instruction/instruction.html">
        
    </a>
</div>
</div>
</body>
<?php
echo "<script> document.getElementById('status').innerHTML = " . $_SESSION['status'] . " "; </script>";
$_SESSION['status'] = "";
?>
</html>

```

singup.php

```

<?php
session_start();
include "database_connect.php";
$login = $_POST['login'];
$password = $_POST['password'];
$check = false;
$admin = false;
$sql = "SELECT id, login, password, type FROM `users`";
$result = mysqli_query($connect, $sql);
if(!$result){
    print(mysqli_error($connect));
}
while($row = mysqli_fetch_array($result)){
    if($row['login'] == $login){
        if($row['password'] == $password){
            $check = true;
            $_SESSION['user'] = $row['id'];
            if($row['type'] == 'admin'){
                $admin = true;
            }
            break;
        }else{
            $_SESSION['status'] = 'невірно введений пароль';
            echo "<script> document.location.href = 'index.php'; </script>";
        }
    }
}

```

```

}
if($check){
    $_SESSION['status'] = "";
    if(!$admin){
        echo "<script> document.location.href = 'staff_page/staff_page.php'; </script>";
    }
    else{
        echo "<script> document.location.href = 'admin_page/admin_page.php'; </script>";
    }
}
else{
    $_SESSION['status'] = 'даний логін не зареєстровано';
    echo "<script> document.location.href = 'index.php'; </script>";
}
mysqli_close($connect);
?>

```

registration.php

```

<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration</title>
    <link rel="stylesheet" href="style.css">
    <link rel="shortcut icon" href="regPic/icon.ico" type="image/x-icon">
    <script src="script.js"></script>
</head>
<body>
    <div class="wrap">
        <div class="reg_window">
            <div class="header">
                <p>
                    реєстрація
                </p>
            </div>
            <form method="POST">
                <p>логін</p>
                <input form="main_form" type="text" name="login" class="login" placeholder="латиниця" required>
                <p>електронна пошта</p>
                <input form="main_form" type="email" name="email" placeholder="електронна@пошта" required>
                <p>пароль</p>
                <input form="main_form" type="password" class="pass" name="password" onchange="check()">
            </form>
        </div>
    </div>

```

```

<button type="button" class="visible" id="closed" onclick="pass(0)">
</button>
<p>повторіть пароль</p>
<input form="main_form" type="password" class="pass" onchange="check()">
<button type="button" class="visible" id="closed" onclick="pass(1)">
</button>
<input form="main_form" type="reset" value="сброс">
<input type="button" value="далі" id="submit" style="grid-column: 2/4; font-size: 20px;" onclick="next()" disabled>
<div class="status">
  <p id="status"></p>
</div>
</form>
</div>
</div>

<!-- *****Second form***** -->

<div class="wrap" style="display: none">
<div class="reg_window" style="height: 510px;">
  <div class="header">
    <p>
      особисті дані
    </p>
  </div>
  <form id="main_form" action="regup.php" method="POST">
    <p>Прізвище</p>
    <input type="text" name="surname" class="surname" required>
    <p>Ім'я</p>
    <input type="text" name="name" class="name" required>
    <p>По батькові</p>
    <input type="text" name="second_name" style="grid-column: 2/4;" class="second_name" required>
    <p>моб. телефон</p>
    <input type="tel" name="phone" class="tel" pattern="[0-9]{3}-[0-9]{2}-[0-9]{2}-[0-9]{3}" placeholder="000-00-00-000" required>
    <p>Підказка</p>
    <input type="text" name="hint" class="hint" placeholder="підказка для паролю">
    <input type="button" value="назад" style="width: 200px; margin-left: 10px; font-size: 20px;" onclick="back()">
    <input type="submit" value="зареєструватися" id="submit1" name="submit1">
  </form>
</div>
</div>
</body>
<?php
echo "<script> document.getElementById('status').innerHTML = " . $_SESSION['status'] . " "; </script>";
$_SESSION['status'] = ";
?>
</html>

```

regup.php

```

<?php
session_start();
include "database_connect.php";
if(!empty($_POST['login'])){// чтоб POST сработал 1 раз
    $phone = $_POST['phone'];
    $login = $_POST['login'];
    $email = $_POST['email'];
    $sql = "SELECT p.phone, u.login, u.email FROM private p, users u";
    $result = mysqli_query($connect, $sql);
    if(!$result){
        $_SESSION['status'] = mysqli_error($connect);
    }
    while($row = mysqli_fetch_array($result)){
        if($row['phone'] == $phone){
            $_SESSION['status'] = 'Вказаний телефон вже зареєстровано';
            echo "<script> document.location.href = 'registration.php'; </script>";
        }
        if($row['login'] == $login){
            $_SESSION['status'] = 'Даний логін вже зареєстровано';
            echo "<script> document.location.href = 'registration.php'; </script>";
        }
        if($row['email'] == $email){
            $_SESSION['status'] = 'Введена електронна пошта вже зареєстрована';
            echo "<script> document.location.href = 'registration.php'; </script>";
        }
    }
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $second = $_POST['second_name'];
    $hint = $_POST['hint'];
    $sql = "SELECT COUNT(id) AS 'count' FROM `private`";
    $result = mysqli_query($connect, $sql);
    if(!$result){
        $_SESSION['status'] = mysqli_error($connect);
    }
    $id = mysqli_fetch_array($result)['count'] + 1;
    $sql = "INSERT INTO `private` (id, `surname`, `name`, `second_name`, `phone`, `hint`) VALUES ('$id', '$surname', '$name', '$second', '$phone', '$hint')";
    $result = mysqli_query($connect, $sql);
    if(!$result){
        $_SESSION['status'] = mysqli_error($connect);
    }
    $password = $_POST['password'];
    $private_id = $id;

```

```

    $sql = "INSERT INTO `users` (`id`, `login`, `email`, `password`, `type`, `private_id`) VALUES ('$id', '$login', '$email', '$password', 'user',
    '$private_id')";
    $result = mysqli_query($connect, $sql);
    if(!$result){
        $_SESSION['status'] = mysqli_error($connect);
    }
    echo "<script> document.location.href = 'index.php'; </script>";
}
mysqli_close($connect);
?>

```

database_connect.php

```

<?php
$servername = "localhost";
$dbname = "flexboard";
$username = "root";
$password = "root";
$connect = mysqli_connect($servername, $username, $password, $dbname);
if(!$connect){
    echo "<script> alert('Connection failed: ' . mysqli_connect_error()) </script>";
}
?>

```

script.js

```

function pass(a){
    if(document.getElementsByClassName("visible")[a].id == "closed"){

        document.getElementsByClassName("visible")[a].id = "open";
        document.getElementsByClassName("pass")[a].type = "text";
    }
    else{
        document.getElementsByClassName("visible")[a].id = "closed";
        document.getElementsByClassName("pass")[a].type = "password";
    }
}

function check(){
    if(document.getElementsByClassName("pass")[0].value != "" && document.getElementsByClassName("pass")[1].value != ""){
        if(document.getElementsByClassName("pass")[0].value.length > 7){
            if(document.getElementsByClassName("pass")[0].value == document.getElementsByClassName("pass")[1].value){
                document.getElementById("submit").disabled = false;
                document.getElementById("status").innerHTML = "";
            }else{
                document.getElementById("submit").disabled = true;
            }
        }
    }
}

```

```

        document.getElementById("status").innerHTML = "Паролі не співпадають. Спробуйте заново";
    }
}
else{
    document.getElementById("submit").disabled = true;
    document.getElementById("status").innerHTML = "Довжина паролю має бути від 8 символів";
}
}
}

function next(){
    document.getElementsByClassName('wrap')[0].style.display = "none";
    document.getElementsByClassName('wrap')[1].style.display = "grid";
}

function back(){
    document.getElementsByClassName('wrap')[1].style.display = "none";
    document.getElementsByClassName('wrap')[0].style.display = "grid";
}

function reg(){
    document.location.href = "registration.php";
}

```

staff_page.php

```

<?php
    session_start();
    include "../database_connect.php";
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <link rel="shortcut icon" href="../regPic/icon.ico" type="image/x-icon">
    <script src="script.js"></script>
    <title>Flexboard</title>
</head>
<body>
<div class="content">
    <div class="wrap">
        <div class="logo">
            <a href="../index.php">Flexboard</a>
        </div>
        <div class="tasks_title">

```

```

    <p>Завдання</p>
</div>
<div class="staff_title">
    <p>Виконавці</p>
</div>
<div class="weekdays">
    <button onclick="ind(0)">
        <p>понеділок</p>
    </button>
    <div class="indicator"></div>
    <button onclick="ind(1)">
        <p>вівторок</p>
    </button>
    <div class="indicator"></div>
    <button onclick="ind(2)">
        <p>середа</p>
    </button>
    <div class="indicator"></div>
    <button onclick="ind(3)">
        <p>четвер</p>
    </button>
    <div class="indicator"></div>
    <button onclick="ind(4)">
        <p>п'ятниця</p>
    </button>
    <div class="indicator"></div>
    <button onclick="ind(5)">
        <p>субота</p>
    </button>
    <div class="indicator"></div>
    <button onclick="ind(6)">
        <p>неділя</p>
    </button>
    <div class="indicator"></div>
    <div class="sign">
        <p>
            Molchanov D.Inc<br>
            IT-81/2
        </p>
        
    </div>
</div>
<div class="tasks" style="display: grid;">
</div>
<?php
$task = 0;

```



```

$today_wday = getdate()['wday'];
$user = $_SESSION['user'];
$start_date = strtotime('-' . $today_wday . ' days');
$start_date = date('Y-m-d', $start_date); //переводим в формат SQL
$end_date = strtotime('+'. 6 - $today_wday . ' days');
$end_date = date('Y-m-d', $end_date);
$sql = "SELECT t.*, COUNT(I1.user_id) as staff, ROUND(IFNULL(I2.count, 0)*100/COUNT(I1.task_id), 0) as percent
      FROM tasks t, list_of_tasks I1
      LEFT JOIN (SELECT task_id, COUNT(status) as 'count' FROM list_of_tasks WHERE status = 'done' GROUP BY task_id) I2 ON I1.task_id
      = I2.task_id
      WHERE t.start_date < '$end_date' AND t.end_date > '$start_date'
      AND I1.task_id = t.id
      GROUP BY I1.task_id";
$result = mysqli_query($connect, $sql);
if(!$result){
    print(mysqli_error($connect));
}
$index = 0;
while($row = mysqli_fetch_array($result)){
    $tasks[$index] = $row;
    $index++;
}
$sql1 = "SELECT l.task_id, u.id, p.sumame, SUBSTR(p.name, 1, 1) as 'name', SUBSTR(p.second_name, 1, 1) as 'second_name', l.status
      FROM list_of_tasks l, users u, private p
      WHERE l.user_id = u.id AND u.private_id = p.id";
$result1 = mysqli_query($connect, $sql1);
if(!$result1){
    print(mysqli_error($connect));
}
$index = 0;
while($row = mysqli_fetch_array($result1)){
    $staff[$index] = $row;
    $index++;
}
$sql2 = "SELECT * FROM `list_of_tasks` WHERE `user_id` = $user";
$result2 = mysqli_query($connect, $sql2);
if(!$result2){
    print(mysqli_error($connect));
}
$index = 0;
while($row = mysqli_fetch_array($result2)){
    $user_tasks[$index] = $row;
    $index++;
}
$start_date = strtotime($start_date);
for($i = 1; $i < 8; $i++){

```

```

echo "<div class='tasks' style='display: none;'>";
$index = 0;
foreach($tasks as $row){

    if($row['id'] == $user_tasks[$index]['task_id']
    && date('Y-m-d', $start_date) >= $row['start_date']
    && date('Y-m-d', $start_date) <= $row['end_date']){
    if($user_tasks[$index]['status'] == "yet" || $user_tasks[$index]['date_of_completion'] >= date('Y-m-d', $start_date)){
        echo "
        <button class='task' onclick='showTaskInfo($task, $i)'>
            <p> " . $row['title'] . " <br>
            " . date('d.m.Y', $start_date) . " | Прогрес - " . $row['percent'] . "% | Виконавці: " . $row['staff'] . "</p>
        </button>
        <div class='task_info'>
            <a href='" . (($row['doc'] != "") ? $row['doc'] . "' target='_blank'" : "#") .
            " class='task_text'>
                <img src='" . $row['pic'] . "'>
            </a>
            <p class='task_text'>
                " . $row['description'] . "
            </p>
            <p class='task_text' id='list'>
                ";
                foreach($staff as $value){
                    if($value['task_id'] == $row['id']){
                        echo $value['surname'] . " " . $value['name'] . ".<br>";
                    }
                }
            </p>
            <p class='task_text' style='margin-left: 10px;'>
                П: " . date('d.m.Y', strtotime($row['start_date'])) . "
            </p>
            <div class='progress'>
                <div class='loadline'>
                    <div class='prog_line' style='width: " . (5*$row['percent']) . "px;'>
                </div>
            </div>
            <div style='margin-top: 1px;'>
                <p class='percent'>" . $row['percent'] . "%</p>
            </div>
            <p class='task_text' style='margin-left: 10px; margin-top: 5px;'>
                К: " . date('d.m.Y', strtotime($row['end_date'])) . "
            </p>
            <form action='done_task.php' method='POST'>

```

```

<button type='submit' class='task_text' name='done_btn' id='done_btn' style = ""
.((($user_tasks[$index]['status'] == 'done') ? 'color: #dfd8c0; background-color: #3c3936;' : 'color: #3c3936; background-color: #dfd8c0;')

">
    виконано
</button>
<input type='hidden' name='task_id' value=" . $row['id'] . ">
<input type='hidden' name='user_id' value='$user'>
<input type='hidden' name='date' value=" . (($user_tasks[$index]['status'] == 'done') ? '0000-00-00' : date('Y-m-d', $start_date)) . ">
<input type='hidden' name='status' value=" . (($user_tasks[$index]['status'] == 'done') ? 'yet' : 'done') . ">
</form>
</div>";
$task++;
}
$index++;
}
}
echo "</div>";
$start_date = strtotime('+1 day', $start_date);
}
?>
<div class="staff">
    <?php
        $sql = "SELECT CONCAT(p.sumame, ' ', SUBSTRING(p.name, 1, 1), '.') as title, CONCAT(p.sumame, ' ', p.name, ' ', p.second_name) as
full_name,
        p.phone, u.email, COUNT(l.task_id) as tasks
        FROM private p, users u, list_of_tasks l
        WHERE p.id = u.private_id
        AND l.user_id = u.id
        AND u.type = 'user'
        GROUP BY l.user_id";
$result = mysqli_query($connect, $sql);
if(!$result){
    print(mysqli_error($connect));
}
$index = 0;
while($row = mysqli_fetch_array($result)){
    echo "
    <button onclick='showStaffInfo($index)'>
    <p> . $row['title'] . "</p>
    </button>
    <div class='staff_info'>
    <p class='staff_text'>
        > " . $row['full_name'] . "
    </p>
    <p class='staff_text'>

```

```

        > ". $row[email] ."
    </p>
    <p class='staff_text'>
        > завдань: ". $row[tasks] ."
    </p>
</div>;

    $index++;
}
?>
</div>
</div>
</body>
</html>

```

done_task.php

```

<?php
include "../database_connect.php";
$status = $_POST['status'];
$date = $_POST['date'];
$task_id = $_POST['task_id'];
$user_id = $_POST['user_id'];
$sql = "UPDATE `list_of_tasks`
    SET `status` = '$status', `date_of_completion` = '$date'
    WHERE `task_id` = '$task_id' AND `user_id` = '$user_id'";
$result = mysqli_query($connect, $sql);
if(!$result){
    print(mysqli_error($connect));
}
echo "<script> document.location.href = 'staff_page.php'; </script>";
?>

```

staff_page/script.js

```

function ind(a){

    document.getElementsByClassName("tasks")[0].style.display = "none";

    for(i=0; i < 7; i++){
        document.getElementsByClassName("indicator")[i].style.backgroundColor = "#ffffff";
        document.getElementsByClassName("tasks")[i+1].style.display = "none";
        document.getElementsByClassName("tasks")[i+1].style.gridAutoRows = "110px";
    }

    document.getElementsByClassName("indicator")[a].style.backgroundColor = "#3c3936";
}

```

```

document.getElementsByClassName("tasks")[a+1].style.display = "grid";

count = Array.from(document.getElementsByClassName("task_info")).length;

for(i = 0; i < count; i++){

    document.getElementsByClassName("task_text")[6*i].style.display = "none";
    document.getElementsByClassName("task_text")[6*i+1].style.display = "none";
    document.getElementsByClassName("task_text")[6*i+2].style.display = "none";
    document.getElementsByClassName("task_text")[6*i+3].style.display = "none";
    document.getElementsByClassName("task_text")[6*i+4].style.display = "none";
    document.getElementsByClassName("task_text")[6*i+5].style.display = "none";

    document.getElementsByClassName("progress")[i].style.display = "none";
    document.getElementsByClassName("task_info")[i].style.height = "0px";
    document.getElementsByClassName("task_info")[i].style.display = "none";
}

}

function showStaffInfo(a){

    if(document.getElementsByClassName("staff_info")[a].style.display === 'none' || document.getElementsByClassName("staff_info")[a].style.display
    == ""){

        document.getElementsByClassName("staff_info")[a].style.display = 'grid';

        setTimeout(
            () => document.getElementsByClassName("staff_info")[a].style.height = '70px', 1);

        setTimeout(
            () => {
                document.getElementsByClassName("staff_text")[3*a].style.display = "block";
                document.getElementsByClassName("staff_text")[3*a+1].style.display = "block";
                document.getElementsByClassName("staff_text")[3*a+2].style.display = "block";
            }, 1000);

    }

    else{

        document.getElementsByClassName("staff_text")[3*a].style.display = "none";
        document.getElementsByClassName("staff_text")[3*a+1].style.display = "none";
        document.getElementsByClassName("staff_text")[3*a+2].style.display = "none";
        document.getElementsByClassName("staff_info")[a].style.height = '0px';

        setTimeout(() => document.getElementsByClassName("staff_info")[a].style.display = 'none',
        1000);
    }
}

```

```

}

}

function showTaskInfo(a, b){

    if(document.getElementsByClassName("task_info")[a].style.display == "none" ||
document.getElementsByClassName("task_info")[a].style.display == ""){

        document.getElementsByClassName("tasks")[b].style.gridAutoRows = "170px";

        document.getElementsByClassName("task_info")[a].style.display = "grid";

        setTimeout(
            () => document.getElementsByClassName("task_info")[a].style.height = "220px", 1);

        setTimeout(
            () => {
                document.getElementsByClassName("task_text")[6*a].style.display = "block";
                document.getElementsByClassName("task_text")[6*a+1].style.display = "block";
                document.getElementsByClassName("task_text")[6*a+2].style.display = "block";
                document.getElementsByClassName("task_text")[6*a+3].style.display = "block";
                document.getElementsByClassName("task_text")[6*a+4].style.display = "block";
                document.getElementsByClassName("task_text")[6*a+5].style.display = "block";

                document.getElementsByClassName("progress")[a].style.display = "grid";
                document.getElementsByClassName("percent")[a].style.display = "block";
            }, 1000);

    }

    else{
        check = true;

        document.getElementsByClassName("task_text")[6*a].style.display = "none";
        document.getElementsByClassName("task_text")[6*a+1].style.display = "none";
        document.getElementsByClassName("task_text")[6*a+2].style.display = "none";
        document.getElementsByClassName("task_text")[6*a+3].style.display = "none";
        document.getElementsByClassName("task_text")[6*a+4].style.display = "none";
        document.getElementsByClassName("task_text")[6*a+5].style.display = "none";

        document.getElementsByClassName("progress")[a].style.display = "none";

        document.getElementsByClassName("task_info")[a].style.height = "0px";

        setTimeout(

```

```

()=> {
    document.getElementsByClassName("task_info")[a].style.display = "none";

    Array.from(document.getElementsByClassName("task_info")).forEach((element) => {
        if(element.style.display != "none"){
            check = false;
        }
    });

    if(check){
        document.getElementsByClassName("tasks")[b].style.gridAutoRows = "110px";
    }

    }, 1000);

}

}

```

admin_page.php

```

<?php
session_start();
include "../database_connect.php";
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <link rel="shortcut icon" href="../regPic/icon.ico" type="image/x-icon">
    <script src="script.js"></script>
    <title>Admin page</title>
</head>
<body>
<div class="content">
    <div class="wrap">
        <div class="logo">
            <a href="../index.php">Flexboard</a>
        </div>
        <div class="tasks_title">
            <p>Завдання</p>
        </div>
        <div class="staff_title">
            <p>Кафедра</p>

```

```

</div>
<div class="weekdays">
  <button onclick="ind(0)">
    <p>додати завдання</p>
  </button>
  <div class="indicator"></div>
  <button onclick="ind(1)">
    <p>редагувати завдання</p>
  </button>
  <div class="indicator"></div>
  <button onclick="ind(2)">
    <p>видалити завдання</p>
  </button>
  <div class="indicator"></div>
  <button onclick="ind(3)">
    <p>виконання</p>
  </button>
  <div class="indicator"></div>
  <button onclick="ind(4)">
    <p>виконавці</p>
  </button>
  <div class="indicator"></div>
  <div class="sign">
    <p>
      Molchanov D.Inc<br>
      IT-81/2
    </p>
    
  </div>
</div>
<div class="tasks" style="display: grid;">
</div>
<div class="tasks" style="display: none;">
  <div class="new_task">
    <form enctype="multipart/form-data" action="add_new_task.php" method="POST" id="new_task_form">
      <p>Заголовок завдання</p>
      <input class="input_title" type="text" name="title" required>
      <p>Текст завдання</p>
      <textarea form="new_task_form" class="input_text" name="text"></textarea>
      <p>Дата початку</p>
      <input type="date" name="start_date" required>
      <p>Дата закінчення</p>
      <input type="date" name="end_date" required>
      <p>Файл документу</p>
      <input type="file" name="file" accept=".docx, .doc, application/pdf">
      <input type="submit" value="створити">
    </form>
  </div>

```



```

<p class="status" style="grid-column: 1/3; text-align: center;">
<?php
    echo $_SESSION['file_status'];
?>
</p>
</form>
</div>
</div>

<div class="tasks" style="display: none;">
<div class="edit_task">
<?php
    $sql = "SELECT * FROM `tasks`";
    $result = mysqli_query($connect, $sql);
    if(!$result){
        print(mysqli_error($connect));
    }
    $index = 0;
    while($row = mysqli_fetch_array($result)){
        echo "
<div class='task'>
<form enctype='multipart/form-data' action='edit_task.php' method='POST' id='edit_form$index'>
    <input type='hidden' name='id' value='". $row['id'] . "'>
    <input type='hidden' name='doc' value='". $row['doc'] . "'>
    <input type='hidden' name='pic' value='". $row['pic'] . "'>
    <p>Заголовок завдання</p>
    <input class='input_title' type='text' name='title' value='". $row['title'] . "'>
    <p>Текст завдання</p>
    <textarea form='edit_form$index' class='input_text' name='description'>". $row['description'] . "</textarea>
    <p>Дата початку</p>
    <input type='date' name='start_date' value='". $row['start_date'] . "'>
    <p>Дата закінчення</p>
    <input type='date' name='end_date' value='". $row['end_date'] . "'>
    <p>Файл документу</p>
    <input type='file' name='file' accept='.docx, .doc, application/pdf'>
    <input type='submit' value='зберегти зміни'>
</form>
</div>";
        $index++;
    }
?>
</div>
</div>
<div class="tasks" style="display: none;">
<div class="delete_task">
<?php

```

```

$sql = "SELECT * FROM `tasks`";
$result = mysqli_query($connect, $sql);
if(!$result){
    print(mysqli_error($connect));
}
$index = 0;
while($row = mysqli_fetch_array($result)){
    echo "
    <div class='task'>
    <form enctype='multipart/form-data' action='delete_task.php' method='POST' id='edit_form$index'>
        <input type='hidden' name='id' value='". $row['id'] . "'>
        <p>Завдання</p>
        <input class='input_title' type='text' name='title' value='". $row['title'] . "' disabled>
        <p>Дата початку</p>
        <input type='date' name='start_date' value='". $row['start_date'] . "' disabled>
        <p>Дата закінчення</p>
        <input type='date' name='end_date' value='". $row['end_date'] . "' disabled>
        <input type='submit' value='Видалити завдання'>
    </form>
    </div>";
    $index++;
}
?>
</div>
</div>
<div class="tasks" style="display: none;">
    <div class="show_task">
        <?php
        $sql = "SELECT `task_id`, COUNT(*) as count, tasks.title
        FROM `list_of_tasks`
        LEFT JOIN tasks ON tasks.id = list_of_tasks.task_id
        GROUP BY `task_id`";
        $result = mysqli_query($connect, $sql);
        $sql1 = "SELECT l.task_id, CONCAT(p.surname, ' ', SUBSTRING(p.name, 1, 1), '.') as name, l.status, l.date_of_completion as date
        FROM list_of_tasks l
        LEFT JOIN users u ON l.user_id = u.id
        LEFT JOIN private p ON u.private_id = p.id
        ORDER BY `T`.`task_id` ASC";
        $result1 = mysqli_query($connect, $sql1);
        $index = 0;
        while($row = mysqli_fetch_array($result1)){
            $staff[$index] = $row;
            $index++;
        }
        $index = 0;
        while($row = mysqli_fetch_array($result)){

```

```

echo "
<div class='task'>
<p>Завдання №" . ($index+1) . "</p>
<input type='text' name='title' value='". $row['title'] . "'>
<p>Виконавці</p>
<div class='staff_list'>;
foreach($staff as $value){
    if($value['task_id'] == $row['task_id']){
        echo "
        <p>" . $value['name'] . "</p>
        <img src='adminPic/" . $value['status'] . ".png'>
        <p>" . (($value['date'] != '0000-00-00') ? date("d.m.Y", strtotime($value['date'])) : " - ") . "</p>;
    }
}
echo "</div>
</div>;
$index++;
}
?>
</div>
</div>
<div class="tasks" style="display: none;">
<div class="assign_task">
<div class="task">
<form action='add_link.php' method='POST' id="add_link_form">
<p>Призначити завдання</p>
<select name='task' form='add_link_form">
<?php
$sql = "SELECT t.id, t.title
        FROM tasks t";
$result = mysqli_query($connect, $sql);
while($value = mysqli_fetch_array($result)){//Вывод всех заданий в список
    echo "
    <option value='". $value['id'] . "'>" . $value['title'] . "</option>;
}
?>
</select>
<p>Виконавці</p>
<select name='user' form='add_link_form">
<?php
$sql = "SELECT u.id, CONCAT(p.surname, ' ', p.name) as name
        FROM users u
        LEFT JOIN private p ON p.id = u.private_id
        WHERE u.type = 'user'";
$result = mysqli_query($connect, $sql);
while($value = mysqli_fetch_array($result)){//Вывод всех исполнителей в список

```

```

        echo "
        <option value=\"" . $value['id'] . ">" . $value['name'] . "</option>";
    }
?>
</select>
<input type="submit" name="submit" value="Призначити">
</form>
</div>
<div class="task">
<form action='remove_link.php' method='POST' id='remove_link_form'>
    <p>Видалити призначення</p>
    <select name='task' form='remove_link_form'>

<?php
$sql = "SELECT t.id, t.title
        FROM tasks t";
$result = mysqli_query($connect, $sql);
while($value = mysqli_fetch_array($result)){//Вывод всех заданий в список
    echo "
        <option value=\"" . $value['id'] . ">" . $value['title'] . "</option>";
    }
?>
</select>
<p>Виконавцю</p>
<select name='user' form='remove_link_form'>

<?php
$sql = "SELECT u.id, CONCAT(p.surname, ' ', p.name) as name
        FROM users u
        LEFT JOIN private p ON p.id = u.private_id
        WHERE u.type = 'user'";
$result = mysqli_query($connect, $sql);
while($value = mysqli_fetch_array($result)){//Вывод всех исполнителей в список
    echo "
        <option value=\"" . $value['id'] . ">" . $value['name'] . "</option>";
    }
?>
</select>
<input type="submit" name="submit" value="Видалити">
</form>
</div>
</div>
</div>
<div class="staff">
<?php
    $sql = "SELECT CONCAT(p.surname, ' ', SUBSTRING(p.name, 1, 1), '.') as title, CONCAT(p.surname, ' ', p.name, ' ', p.second_name) as
full_name,

```

```

        p.phone, u.email, COUNT(l.task_id) as tasks
    FROM private p, users u, list_of_tasks l
    WHERE p.id = u.private_id
    AND l.user_id = u.id
    AND u.type = 'user'
    GROUP BY l.user_id";
$result = mysqli_query($connect, $sql);
if(!$result){
    print(mysqli_error($connect));
}
$index = 0;
while($row = mysqli_fetch_array($result)){

    echo "
    <button onclick='showStaffInfo($index)'>
    <p>" . $row['title'] . "</p>
    </button>
    <div class='staff_info'>
        <p class='staff_text'>
            > " . $row['full_name'] . "
        </p>
        <p class='staff_text'>
            > " . $row['email'] . "
        </p>
        <p class='staff_text'>
            > завдань: " . $row['tasks'] . "
        </p>
    </div>";
    $index++;
}
$_SESSION['file_status'] = "
?>
</div>
</div>
</div>
</body>
</html>

```

add_link.php

```

<?php
include "../database_connect.php";
$sql = "SELECT COUNT(*) as count FROM `list_of_tasks`";
$result = mysqli_query($connect, $sql);
if(!$result){
    print(mysqli_error($connect));
}

```

```

Sid = (mysqli_fetch_array($result))['count'] + 1;
$task = $_POST['task'];
$user = $_POST['user'];
$sql = "INSERT INTO `list_of_tasks` (`id`, `task_id`, `user_id`, `status`, `date_of_completion`)
VALUES ('$id', '$task', '$user', 'yet', '0000-00-00')";
$result = mysqli_query($connect, $sql);
echo "<script> document.location.href = 'admin_page.php'; </script>";
?>

```

add_new_task.php

```

<?php
session_start();

if($_POST['end_date'] < $_POST['start_date']){
    $_SESSION['file_status'] = "Невірно введена дата";
    echo "<script> document.location.href = 'admin_page.php'; </script>";
    die();
}
if(!empty($_FILES['file']['name'])){
    switch($_FILES['file']['error']){//перевірка помилок
        case 0:
            $_SESSION['file_status'] = "Завдання успішно створено";
            break;
        case 1:
            $_SESSION['file_status'] = "Розмір прийнятого файлу перевищив максимально допустимий розмір, властивий системі.
Максимальний розмір - 2Мб";
            break;
        case 2:
            $_SESSION['file_status'] = "Розмір файлу, що завантажується, перевищив максимально допустимий розмір. Максимальний розмір -
2Мб";
            break;
        case 3:
            $_SESSION['file_status'] = "Завантажуваний файл було отримано лише частково";
            break;
        case 4:
            $_SESSION['file_status'] = "Файл не було завантажено";
            break;
        case 6:
            $_SESSION['file_status'] = "Відсутня тимчасова папка";
            break;
        case 7:
            $_SESSION['file_status'] = "Не вдалося записати файл";
            break;
        case 8:
            $_SESSION['file_status'] = "Модуль PHP зупинив завантаження файлу";
            break;

```

```

}
if($_FILES['file']['error'] != 0){
    echo "<script> document.location.href = 'admin_page.php'; </script>";
    die();
}
$uploadway = "../staff_page/docs/" . basename($_FILES['file']['name']); //путь к файлу
move_uploaded_file($_FILES['file']['tmp_name'], $uploadway); //перемещение файла в директорию
}
include "../database_connect.php";
$sql = "SELECT COUNT(*) as count FROM `tasks`"; //Считаем количество записей
$result = mysqli_query($connect, $sql);
if(!$result){
    $_SESSION['file_status'] = mysqli_error($connect);
}

$id = (mysqli_fetch_array($result))['count'] + 1; //аиди новой записи
$title = $_POST['title'];
$description = $_POST['text'];
$start_date = date("Y-m-d", strtotime($_POST['start_date']));
$end_date = date("Y-m-d", strtotime($_POST['end_date']));
if(!empty($_FILES['file']['name'])){
    $pic = 'staffPic/2.png';
    $doc = "docs/" . $_FILES['file']['name'];
}else{
    $pic = 'staffPic/3.png';
    $doc = "";
}
$sql = "INSERT INTO `tasks` (`id`, `title`, `description`, `start_date`, `end_date`, `pic`, `doc`)
VALUES ($id, '$title', '$description', '$start_date', '$end_date', '$pic', '$doc')"; //вставка нового задания
$result = mysqli_query($connect, $sql);
if(!$result){
    $_SESSION['file_status'] = mysqli_error($connect);
}else{
    $_SESSION['file_status'] = "Завдання успішно створено";
}
echo "<script> document.location.href = 'admin_page.php'; </script>";
?>

```

delete_task.php

```

<?php
session_start();
include "../database_connect.php";
$id = $_POST['id'];
$sql = "DELETE FROM `tasks` WHERE `tasks`.`id` = $id";
$result = mysqli_query($connect, $sql);
if($result){

```

```

$sql = "SELECT COUNT(*) as count FROM `tasks` WHERE id > $id";
$result = mysqli_query($connect, $sql);
$count = mysqli_fetch_array($result)['count'];
$sql = "";
for($i = 0; $i < $count; $i++, $id++){
    $sql .= "UPDATE `tasks` SET `id` = '$id' WHERE `tasks`.`id` = " . ($id+1) . " . ";
}
$result = mysqli_multi_query($connect, $sql);
}
echo "<script> document.location.href = 'admin_page.php'; </script>";
?>

```

edit_task.php

```

<?php
include "../database_connect.php";
if($_POST['end_date'] < $_POST['start_date']){
    echo "<script> document.location.href = 'admin_page.php'; </script>";
    die();
}
if(!empty($_FILES['file']['name'])){
    if($_FILES['file']['error'] != 0){
        echo "<script> document.location.href = 'admin_page.php'; </script>";
        die();
    }
    if($_POST['doc'] != ""){
        unlink("../staff_page/" . $_POST['doc']);
    }
    $uploadway = "../staff_page/docs/" . basename($_FILES['file']['name']); //путь к файлу
    move_uploaded_file($_FILES['file']['tmp_name'], $uploadway); //перемещение файла в директорию
}
$id = $_POST['id'];
$title = $_POST['title'];
$description = $_POST['description'];
$start_date = date("Y-m-d", strtotime($_POST['start_date']));
$end_date = date("Y-m-d", strtotime($_POST['end_date']));
if(!empty($_FILES['file']['name']) && $_POST['pic'] == "staffPic/3.png"){
    $pic = "staffPic/2.png";
}else{
    $pic = $_POST['pic'];
}
if(!empty($_FILES['file']['name'])){
    $doc = "docs/" . $_FILES['file']['name'];
}else{
    $doc = $_POST['doc'];
}
$sql="UPDATE `tasks`

```



```

SET `title` = '$title',
`description` = '$description',
`start_date` = '$start_date',
`end_date` = '$end_date',
`pic` = '$pic',
`doc` = '$doc'
WHERE `tasks`.`id` = $id";
$result = mysqli_query($connect, $sql);
echo "<script> document.location.href = 'admin_page.php'; </script>";
?>

```

remove_link.php

```

<?php

include "../database_connect.php";
$task = $_POST['task'];
$user = $_POST['user'];
$sql = "SELECT id
      FROM `list_of_tasks`
      WHERE `list_of_tasks`.`task_id` = '$task'
      AND `list_of_tasks`.`user_id` = '$user'";
$result = mysqli_query($connect, $sql);
$id = mysqli_fetch_array($result)['id'];
$sql = "DELETE FROM `list_of_tasks`
      WHERE `list_of_tasks`.`id` = '$id'";
$result = mysqli_query($connect, $sql);
if($result){
    $sql = "SELECT COUNT(*) as count FROM `list_of_tasks` WHERE id > $id";
    $result = mysqli_query($connect, $sql);
    $count = mysqli_fetch_array($result)['count'];
    $sql = "";
    for($i = 0; $i < $count; $i++, $id++){
        $sql .= "UPDATE `list_of_tasks` SET `id` = '$id' WHERE `list_of_tasks`.`id` = " . ($id+1) . " . ";
    }
    $result = mysqli_multi_query($connect, $sql);
}
echo "<script> document.location.href = 'admin_page.php'; </script>";
?>

```

admin_page/script.js

```

function ind(a){
    document.getElementsByClassName("tasks")[0].style.display = "none";
    for(i = 0; i < 5; i++){
        document.getElementsByClassName("indicator")[i].style.backgroundColor = "#ffffff";
        document.getElementsByClassName("tasks")[i+1].style.display = "none";
    }
}

```

```

    document.getElementsByClassName("tasks")[i+1].style.gridAutoRows = "110px";
}
document.getElementsByClassName("indicator")[a].style.backgroundColor = "#3c3936";
document.getElementsByClassName("tasks")[a+1].style.display = "grid";
}

function showStaffInfo(a){
    if(document.getElementsByClassName("staff_info")[a].style.display === 'none' || document.getElementsByClassName("staff_info")[a].style.display
=== ""){
        document.getElementsByClassName("staff_info")[a].style.display = 'grid';
        setTimeout(
            () => document.getElementsByClassName("staff_info")[a].style.height = '70px', 1);
        setTimeout(
            () => {
                document.getElementsByClassName("staff_text")[3*a].style.display = "block";
                document.getElementsByClassName("staff_text")[3*a+1].style.display = "block";
                document.getElementsByClassName("staff_text")[3*a+2].style.display = "block";
            }, 1000);
        }
    else{
        document.getElementsByClassName("staff_text")[3*a].style.display = "none";
        document.getElementsByClassName("staff_text")[3*a+1].style.display = "none";
        document.getElementsByClassName("staff_text")[3*a+2].style.display = "none";
        document.getElementsByClassName("staff_info")[a].style.height = '0px';
        setTimeout(() => document.getElementsByClassName("staff_info")[a].style.display = 'none',
            1000);
        }
    }
}

```

instructions.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <link rel="shortcut icon" href="./regPic/icon.ico" type="image/x-icon">
    <script src="script.js"></script>
    <title>Flexboard</title>
</head>

<body>

<div class="content">
    <div class="wrap">

```

```
<div class="logo">
  <a href="../index.php">Flexboard</a>
</div>
```

```
<div class="tasks_title">
  <p>Інструкція</p>
</div>
```

```
<div class="weekdays">
  <button onclick="ind(0)">
    <p>Створення завдання</p>
  </button>
  <div class="indicator"></div>
```

```
<button onclick="ind(1)">
  <p>Редагування завдання</p>
</button>
<div class="indicator"></div>
```

```
<button onclick="ind(2)">
  <p>Видалення завдання</p>
</button>
<div class="indicator"></div>
```

```
<button onclick="ind(3)">
  <p>Призначення завдань</p>
</button>
<div class="indicator"></div>
```

```
<button onclick="ind(4)">
  <p>реєстрація</p>
</button>
<div class="indicator"></div>
```

```
<button onclick="ind(5)">
  <p>авторизація</p>
</button>
<div class="indicator"></div>
```

```
<button onclick="ind(6)">
  <p>контроль виконання</p>
</button>
<div class="indicator"></div>
```

```
<div class="sign">
  <p>
```

Molchanov D.Inc

IT-81/2

</p>

</div>

</div>

<div class="tasks" style="display: grid;">

</div>

<div class="tasks" style="display: none">

<p>

Для того, щоб створити нове завдання, необхідно авторизуватися як завідуючий(-ча) кафедри та натиснути кнопку "Створення завдання".

</p>

<p>

Після заповнення полів відповідними даними, за необхідності, переходимо до прикріплення файлу.

</p>

<p>

Для прикріплення документа необхідно натиснути на кнопку вибору файлу та обрати .pdf або .doc/.docx файл.

Розмір обраного документа не повинен перевищувати 2 мб.

</p>

<p>

Для збереження усієї введеної інформації та прикріпленого (за потреби) файлу у нове завдання, необхідно натиснути кнопку "Створити".

</p>

<p>

Повідомлення про статус завдання буде показано в нижній частині форми.

</p>

</div>

<div class="tasks" style="display: none;">

<p>

Для того, щоб відредагувати існуюче завдання, необхідно авторизуватися як завідуючий(-ча) кафедри та натиснути кнопку "редагування завдання".

Для редагування, в базі даних повинно бути створене хоча б одне завдання.

</p>

<p>

Всі створені завдання виведені в блоках у три рядки.

Після знаходження необхідного завдання, є можливість відредагувати наведені текстові поля.

</p>

<p>

Для прикріплення нового документу необхідно натиснути на кнопку вибору файлу та обрати .pdf або .doc/.docx файл.

Розмір обраного документа не повинен перевищувати 2 мб.

У базі даних зміниться інформація про прикріплений файл.

Минулий документ буде видалено.

</p>

<p>

Для збереження усієї відредагованої інформації та прикріпленого (за потреби) файлу у існуюче завдання, необхідно натиснути кнопку "Зберегти зміни".

Якщо під час редагування не відбулася зміна файлу, то існуючий документ залишається прикріпленим до завдання.

</p>

<p>

Після перезавантаження сторінки, відредагована інформація буде показана у відповідному блоці завдання.

</p>

</div>

<div class="tasks" style="display: none;">

<p>

Для того, щоб видалити існуюче завдання, необхідно авторизуватися як завідуючий(-ча) кафедри та натиснути кнопку "Видалення завдання".

Для видалення, в базі даних повинно бути створене хоча б одне завдання.

</p>

<p>

Всі створені завдання виведені в блоках у три рядки.

Після знаходження необхідного завдання, є можливість його видалити за допомогою натиснення кнопки "Видалити завдання".

</p>

<p>

Оновлений список завдань буде наведений у всіх вкладках панелі адміністратора.

</p>

</div>

<div class="tasks" style="display: none;">

<p>

Для перевірки усіх призначених завдань необхідно авторизуватися як завідуючий(-ча) кафедри та натиснути кнопку "Виконавці".

</p>

<p>

Щоб надати завданню нового виконавця необхідно натиснути на кнопку "Виконавці".

Для процесу призначення необхідно створити хоча б одне завдання та зареєструвати хоча б одного користувача.

</p>

<p>

Обираємо необхідне завдання з списку всіх створених.

</p>

<p>

Обираємо відповідного виконавця, якому буде призначене нове завдання.

</p>

<p>

Для створення нового призначення, після обрання потрібних варіантів, натискаємо кнопку "Призначити".

</p>

<p>

Новостворене призначення буде відображене на вкладці "Виконання".

Кожному завданню відведений свій блок та список виконавців, поряд з якими зображено їх статус та дату виконання завдання.

</p>

<p>

Для видалення призначення необхідно перейти до вікна "Виконавці", обрати дані для видалення та натиснути кнопку "Видалити".

</p>

<p>

Усі проведені зміни відобразатимуться на вкладці "Виконання".

</p>

```
</div>
```

```
<div class="tasks" style="display: none;">
```

```

```

```
<p>
```

Для реєстрації нового користувача необхідно на формі авторизації натиснути кнопку "Реєстрація".

```
</p>
```

```

```

```
<p>
```

Після введення у відповідні форми коректних даних необхідно натиснути кнопку "Далі".

Кнопка переходу не буде активною, доки пароль не буде введено за вимогами.

Для оновлення досяжності кнопки, необхідно після змін з паролями клікнути за межами форми.

```
</p>
```

```

```

```
<p>
```

На вкладці "Особисті дані" необхідно ввести інформацію про персону користувача та натиснути "Зареєструватися".

В разі виникнення помилок, інформація буде виведена у статусну строку в самому кінці форми.

```
</p>
```

```
</div>
```

```
<div class="tasks" style="display: none;">
```

```

```

```
<p>
```

Після введення у відповідні форми коректних даних необхідно натиснути кнопку "Авторизуватися".

Кнопка авторизації не буде активною, доки пароль не буде введено за вимогами.

Після успішної авторизації, користувач буде перенаправлений на сторінку, відповідну його статусу (інформаційне поле або панель адміністратора).

```
</p>
```

```

```

```
<p>
```

В разі порушення вимог, текст помилки буде виведено в кінці форми.

Для оновлення досяжності кнопки, необхідно після змін з паролями клікнути за межами форми.

```
</p>
```

```
</div>
```

```
<div class="tasks" style="display: none;">
```

```

```

```
<p>
```

Для взаємодії за призначеним завданням, необхідно авторизуватися як працівник кафедри (user).

```
</p>
```

```

```

```
<p>
```

Щоб подачити завдання, необхідно обрати поточний день тижня натиснувши кнопку з відповідною назвою, після чого на екран виведеться весь список завдань, термін дії яких припадає на обраний день тижня.

</p>

<p>

Для отримання повної інформації про завдання, необхідно натиснути на нього.

Панель з додатковою інформацією та елементами взаємодії зв'яжеться знизу обраного завдання.

</p>

<p>

Для взаємодії, користувачу доступна кнопка "Виконано", яка знаходиться посередині знизу форми та прикріплений документ (за наявності).

Щоб відкрити прикріплений документ, необхідно натиснути на значок документів, зліва від повного опису завдання.

Якщо до завдання документ не прикріплено, то натискання на відповідний значок нічого не дасть.

</p>

<p>

Після натиснення кнопки "Виконано", в базу даних запишеться дата виконання завдання та змінеться його статус.

Зміни в базі даних напряму відобразатимуться у описі завдання: зміниться відсоток та лінія прогресу виконання, а також завдання не буде відображатися в наступні дні, після виконання.

Ще одне натиснення поверне завдання до статусу невиконаного.

</p>

</div>

</div>

</div>

</body>

</html>

instructions/script.js

```
function ind(a){
  document.getElementsByClassName("tasks")[0].style.display = "none";
  for(i = 0; i < 7; i++){
    document.getElementsByClassName("indicator")[i].style.backgroundColor = "#ffffff";
    document.getElementsByClassName("tasks")[i+1].style.display = "none";
  }
  document.getElementsByClassName("indicator")[a].style.backgroundColor = "#3c3936";
  document.getElementsByClassName("tasks")[a+1].style.display = "grid";
}
```