Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

РОЗРОБКА ВЕБ-ДОДАТКУ З ПЛАНУВАННЯ ЧАСУ

Текстова частина до курсової роботи

за спеціальністю «Інженерія програмного забезпечення» 121

Керівник курсової роботи

Ст.викладач Борозенний С.О.

_____

(підпис)

“___” _____ 2022 р.

Виконав студент Кучерявий В. Ю.

“___” _____ 2022 р.

Київ 2022

ЗМІСТ

АНОТАЦІЯ

В процесі виконання роботи відбулося більш детальне ознайомлення із поняттям тайм-менеджменту, як він впливає на продуктивність людини, для чого він потрібен і було розроблено додаток, який допомагає користувачеві контролювати свій час, пріорітезувати задачі, слідкувати за своєю продуктивністю.

Проведено аналіз сфери тайм-менеджменту в цілому, огляд існуючих аналогів на ринку та актуальність використання веб-застосунків, , на основі цих досліджень виділено сильні та слабкі сторони.

Ключові слова: тайм-менеджмент, планування, аналіз, розробка, Python, Django, веб-додаток, мова програмування, фреймворк.

ВСТУП

Актуальність дослідження зумовлена стрімким розвитком використання засобів автоматизації, таких як електронні системи обробки заявок в тій чи іншій тематиці, автоматизацією усіх процесів життєдіяльності людини.

Одним з прикладів системи обробки заявок можна вважати різного роду планувальники часу, які допомагають створити ряд заявок (для себе, або для команди) і слідкувати за їх виконанням.

Виходячи з високої актуальності теми створення різного роду планувальників, об’єктом дослідження даної роботи є процес контролю часу з використанням програмних продуктів.

Предметом дослідження є веб-додатки для планування часу.

Основною метою роботи є створення власної реалізації веб-додатку для планування часу.

Для досягнення поставленої мети слід виконати наступні завдання:

- провести аналіз поняття тайм-менеджменту;
- провести аналіз поняття веб-додатку;
- провести аналіз поняття веб-розробки;
- провести огляд мови програмування;
- провести огляд середовища розробки;
- розробити методи та компоненти системи;
- розробити алгоритми і моделі;
- розробити внутрішню структуру
- розробити графічний інтерфейс користувача;

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Тайм-менеджмент

Тайм-менеджмент - це процес навмисного розподілу часу ефективно та продуктивно. Це включає в себе визначення пріоритетів завдань і обов'язків, а потім планування тижня, щоб переконатися, що ваш час витрачається на найважливіші роботи. Ефективно плануючи те, як ви будете витрачати кожен день, можна звести до мінімуму відволікання і забезпечите, щоб найбільша частина часу була витрачена на найважливіші завдання.

Зрештою, у стандартному робочому дні всього стільки годин. Ефективне управління часом дозволяє нам розумно вибирати, як ми будемо їх використовувати, і, у свою чергу, завдання, на які не вимагають виділення нашого обмеженого часу.

У сучасному світі постійної роботи багато людей думають, що виглядати зайнятим є ключем до успіху. Деякі люди вважають успішною людиною того, хто постійно поспішає від однієї зустрічі до іншої, їсть обід за столом, одночасно підключаючись до конференц-дзвінка та перевіряючи електронну пошту.

Але насправді це далеко не те, як насправді виглядає «успіх». Натомість це класичний випадок людини з поганим управлінням часом.

Людина, яка є ефективним менеджером свого часу, зустрічається зовсім по-різному. Зазвичай вони вважаються надійними, продуктивними та здатними дотримуватись своїх термінів. Якість їхньої роботи стабільно висока і вони мають гарну професійну репутацію. Вони також мають успішну кар'єру, а їхні досягнення винагороджуються.

Люди, які не мають часу, часто мають одну спільну рису: замість того, щоб вирішувати, які завдання є найважливішими та терміновими, і, таким чином, виділяють відповідну кількість часу, щоб забезпечити їх виконання відповідно до стандартів якості, вони проводять більшу частину свого дня,

відповідаючи на кожне. прохання, яке приходить їм на шляху. В результаті вони витрачають найменшу кількість часу на виконання своїх найважливіших завдань, тому що вони занадто зайняті тим, що відволікається на те, що потрапило до їхньої поштової скриньки.

1.2 Опис існуючих аналогів для планування часу

На сьогоднішній день на ринку існує доволі багато веб-застосунків, які надають можливість контролювати свій час. Для того щоб усвідомити, що саме повинен містити мій застосунок, то потрібно розглянути існуючі додатки. Для прикладу візьмемо три веб-застосунки: Trello, Jira та Asana.

Trello – це візуальний інструмент для управління роботою, який дозволяє командам обмірковувати, планувати та вести спільну роботу, а також відзначати успіхи. З ним команда буде згуртованою, продуктивною та організованою.

Рисунок 1.2.1 - Trello

Trello підлаштовується під будь-які умови незалежно від того, починаєте ви та ваша команда новий проект або намагаєтеся краще організувати поточну роботу. Цей інструмент дозволить спростити та уніфікувати робочий процес команди інтуїтивно зрозумілим способом. Проте простота Trello приховує великі здібності. Інструмент легко освоїти, але при цьому за допомогою його можна вести найскладніші проекти.

Цей інструмент надає багато функціоналу, основний з всього:

1. Створення проекту
2. Додавання задач та завдань
3. Можливість додавати інших людей до проекту

4. Налаштування прав доступу для різних користувачів
5. Можливість одночасної взаємодії багатьох людей

Тому підсумовуючі можливості Trello, це зручне та гнучке онлайн-рішення для швидкої та ефективної організації роботи, яке однаково добре підходить як для невеликих команд, так і для корпорацій, де може використовуватись для управління окремими проектами або відстеження процесу виконання завдань усередині різних підрозділів.

Перейдемо до іншого доступного аналогу. Jira – це програмний інструмент управління проектами, розроблений компанією Atlassian. Jira часто використовується в IT-компаніях для формування списку завдань, відстеження загального прогресу команди та вирішення проблем, що виникають у процесі розробки продукту.

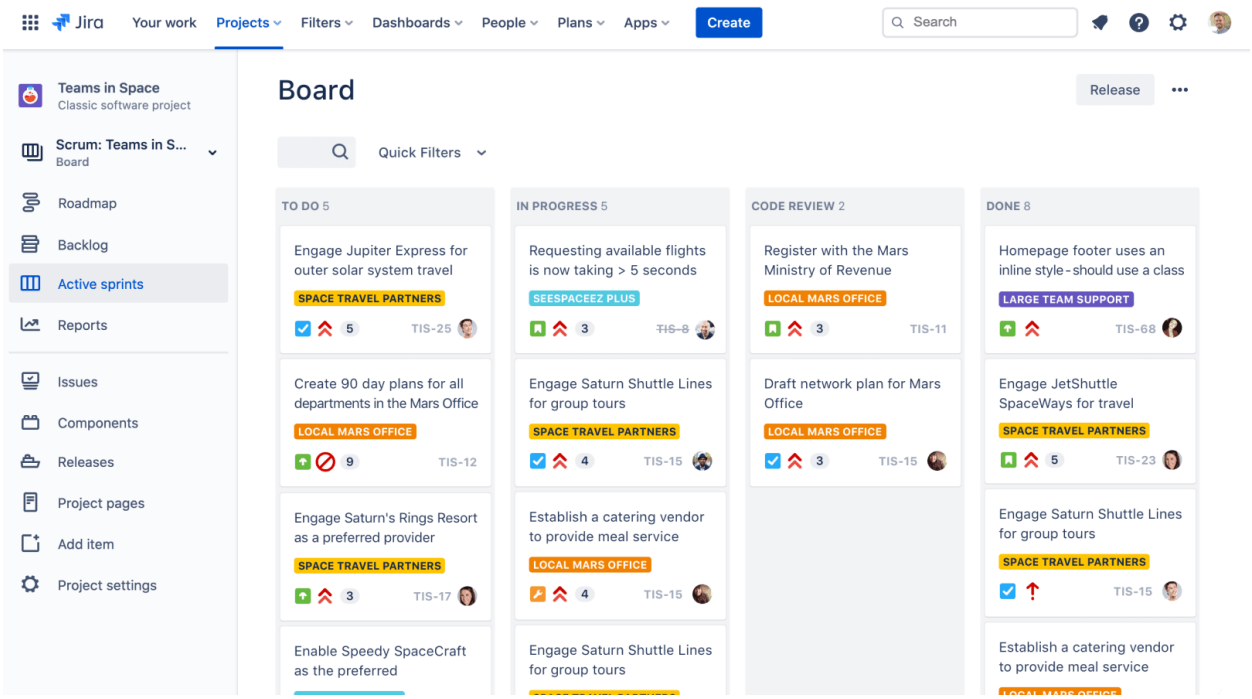


Рисунок 1.2.2 - Jira

Програма Atlassian побудована за принципами канбан-і скрам-дошок, давньої практики організації завдань. Але ці принципи доповнюються масою допоміжних механізмів, які додавалися в додаток виключно з метою спростити створення нових додатків, додати до них функції, виправити помилки тощо. Також ця система управління проектами сповідує Agile-методику розробки.

За допомогою Jira можна легко організувати такі процеси:

1. Наочна організація переліку завдань.
2. Управління проектом та командою, що займається його розвитком.
3. Розробка програмного забезпечення з нуля або додавання нових функцій.
4. Управління завданнями, пов'язаними з маркетинговою складовою продукту.
5. Відстеження помилок у програмі та їх своєчасне виправлення.

Перейдемо до останнього аналогу, який вирішує проблеми тайм-менеджменту.

Asana — це настроювана система керування вмістом робочого місця (CMS), яка розроблена, щоб допомогти широкому колу компаній досягти своїх організаційних потреб. Простіше кажучи, це просунутий організаційний інструмент, який допомагає оптимізувати проекти.

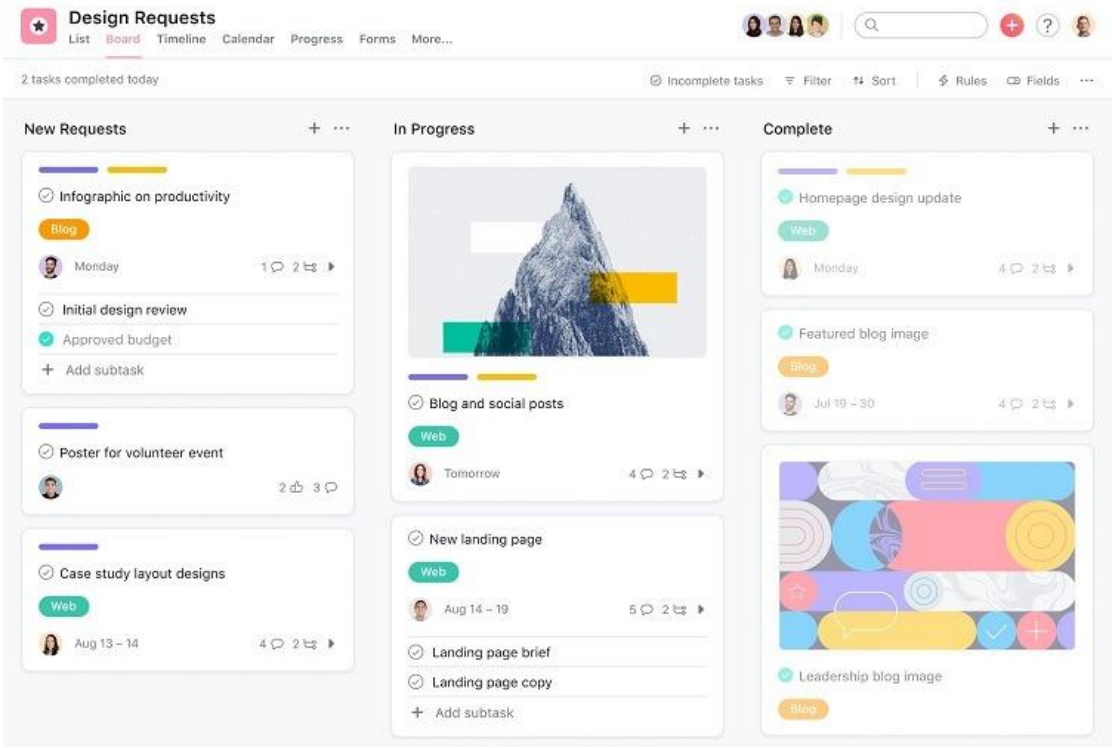


Рисунок 1.2.3 - Asana

Головною особливістю Asana є інструменти управління проектами та завданнями.За допомогою Asana можна створювати списки чи дошки для

розміщення певних проектів у цілому — вони детально описують усі ініціативи, зустрічі та програми, залучені до цих проектів.

Також є можливим розділити ці проекти на завдання та підзавдання, щоб зробити їх більш керованими, і перерахувати кроки для їх виконання. Основний функціонал:

1. Конкретні завдання можна розподілити між конкретними людьми, що було зрозуміло хто над чим працює
2. Є можливим призначити дату початку та терміни виконання
3. Можливість додавання залежностей завдань, щоб указати, коли одні завдання потрібно виконати, перш ніж почати інші
4. Перегляд своїх завдання на годинній шкалі, щоб усі могли бачити проект у хронологічному порядку

Розглянувши обидва додатки можна прийти до висновку, що в загальному ці застосунки об’єднує можливість створення проектів, взаємодії людей в середині додатку та розподілу задач в середині. Але, на мою думку, оскільки в загальному поняття тайм-менеджмент несе за собой можливість власного контролю над своїм часом, тому дуже слушним буде включити в розробку додатку також й статистку, яка допоможе людини організувати свій час та зрозуміти як він рухається в загальному, скільки задач він виконав за якийсь період часу, чи не впала його продуктивність, тощо. Тому це буду одною з основних функцій розробки.

1.3 Поняття веб-додатку та веб-розробки

Веб-додаток (Web-app) — це прикладна програма, яка зберігається на віддаленому сервері та доставляється через Інтернет через інтерфейс браузера. Веб-сервіси за визначенням є веб-додатками, і багато веб-сайтів, хоча й не всі, містять веб-програми. Будь-який компонент веб-сайту, який виконує певну функцію для користувача, кваліфікується як веб-програма.

Веб-додатки можуть бути розроблені для широкого спектру використання і можуть використовуватися будь-ким; від організації до окремої особи з багатьох причин. Зазвичай використовувані веб-додатки можуть включати веб-пошту, онлайн-калькулятори або магазини електронної комерції. До деяких веб-програм можна отримати доступ лише через певний браузер; однак більшість із них доступні незалежно від браузера.

Веб-програми не потрібно завантажувати, оскільки доступ до них здійснюється через мережу. Користувачі можуть отримати доступ до веб-програми через веб-браузер, такий як Google Chrome, Mozilla Firefox або Safari.

Щоб веб-програма працювала, їй потрібні веб-сервер, сервер додатків і база даних. Веб-сервери керують запитами, які надходять від клієнта, а сервер додатків виконує запитане завдання. Базу даних можна використовувати для зберігання будь-якої необхідної інформації.

Веб-додатки зазвичай мають короткі цикли розробки і можуть бути створені невеликими командами розробників. Більшість веб-програм написані на JavaScript, HTML5 або каскадних таблицях стилів (CSS). Програмування на стороні клієнта зазвичай використовує ці мови, які допомагають створювати інтерфейс програми. Програмування на стороні сервера виконується для створення сценаріїв, які використовуватиме веб-додаток. Такі мови, як Python, Java і Ruby, зазвичай використовуються в програмуванні на стороні сервера.

Для повного розуміння, чому було обрано саме веб-застосунок, а не наприклад десктопна версія застосунку, то потрібно ознайомитись с плюсами які привносить саме розробка веб-застосунку.

Веб-додатки мають багато різних застосувань, і разом із цим використанням приходить багато потенційних переваг. Деякі загальні переваги веб-програм:

1. Надання доступу кільком користувачам до однієї версії програми.
2. Веб-програми не потрібно встановлювати.
3. Доступ до веб-програм можна отримати через різні платформи, такі як комп’ютер, ноутбук або мобільний.
4. Можна отримати доступ через кілька браузерів.
5. Швидкий доступ до веб-ресурсу

У секторі мобільних обчислень веб-програми іноді протиставляють рідним додаткам, які є програмами, розробленими спеціально для певної платформи або пристрою та встановленими на цьому пристрої. Однак ці два не виключають один одного. Нативні програми – це програми, які зазвичай завантажуються та створюються спеціально для типу пристрою, на який вони завантажуються. Нативні програми зазвичай можуть використовувати спеціальне обладнання для пристрою, наприклад GPS або камеру в мобільному рідному додатку.

Програми, що поєднують обидва підходи, іноді називають гібридними додатками. Гібридні програми працюють подібно до веб-додатків, але встановлюються на пристрої так само, як і рідні програми. Гібридні додатки також можуть використовувати ресурси, що стосуються пристрою, використовуючи внутрішні API. Завантажені рідні програми іноді можуть працювати в автономному режимі; однак гібридні програми не мають цієї функції. Гібридні додатки зазвичай мають подібні елементи навігації, як і веб-програми, оскільки вони засновані на веб-програмах.

Проаналізувавши плюси веб-розробки, та в загальному всю її специфіку, можна прийти до підсумку, що для реалізації програми з контролю власного часу прекрасно підійде саме веб-застосунок, оскільки він надає можливість користувачам одночасно взаємодіяти у додатку, робити невідкладні зміні,

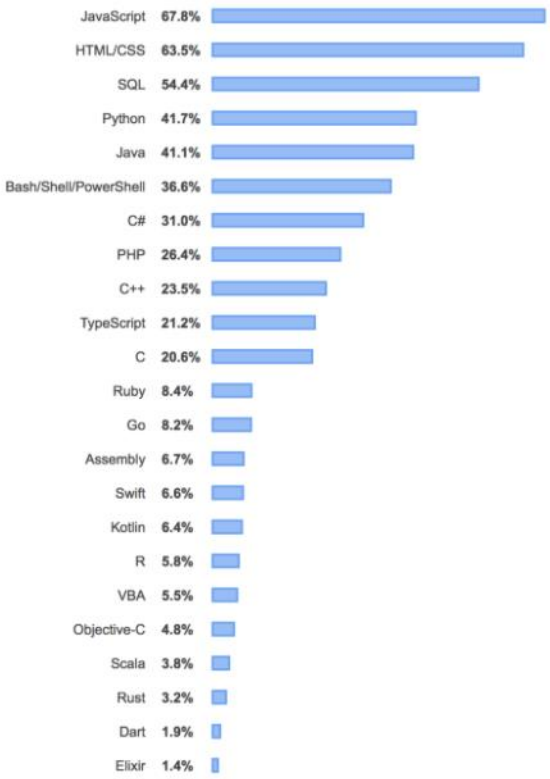
вносити корегування, створювати та оновлювати статуси задач, коментувати свої наробітки в рамках проекту, або певної поставленої задачі.

РОЗДІЛ 2
АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ

2.1 Огляд обраних мов програмування для реалізації фронт-енду та бек-енду та використаних фреймворків

Перед розробкою будь-якого програмного забезпечення потрібно завчасно обрати весь перелік функціоналу, який знадобиться для реалізації проекту, оскільки правильно обрана мова програмування та всі потрібні бібліотеки значно полегшують життя розробнику.

У розробці програмного забезпечення, якщо нам потрібно вибрати мову для проекту, ми повинні поставити собі кілька запитань, перш ніж приймати рішення. Наприклад, що це за проект? Масштабованість програми, складність програми, бюджет розробки, ліміт часу розробки, безпека програми, доступні ресурси тощо. Команда проекту завжди хоче, щоб програма залишалася на тривалий час і відповідала потребам клієнта, навіть якщо бізнес зміни відбуваються пізніше.



На цьому рисунку зображені результати дослідження компанії StackOverflow. Вони ілюструють наскільки часто та чи інша згадана у запитанні на їх сайті. Можемо зробити з цього невеличкий підсумок, що невід'ємної частиною багатьох проектів з реалізації фронт-енду є зв'язка: HTML, CSS та Javascript.

Чому ж взагалі самі ці три основні технології є найпопулярнішими допоможе розібратися більш глибоке занурення у дослідження.

По суті, HTML використовується для створення основного вмісту веб-сторінки, надання їй структури. Починаючи написання слів, а потім застосування до цих слів теги або елементів. Потім веб-браузер зчитує це і може зрозуміти заголовок сторінки, будь-які абзаци, а також те, де починається та закінчується сторінка, таким чином наповнюючи вашу веб-сторінку вмістом. HTML підтримується кожним браузером і встановлюється практично на кожній існуючій веб-сторінці. Вам не потрібні ліцензії, вам не потрібно за це платити, і це може бути досить легко навчитися та кодувати. Якщо ми можемо порівняти веб-сторінку з людським тілом, то HTML — це кістки тіла.

Якщо HTML - це кістки тіла, то CSS - це шкіра, яка його покриває. Він використовується для кольору фону, стилю, макета, кордонів, затінення – усіх основних елементів дизайну, які роблять веб-сторінку гладкою та елегантною. CSS дозволяє розрізняти презентацію та вміст, змінюючи дизайн та відображення елементів HTML. Простота використання — це основна річь, яку CSS придбав у веб-дизайні, трансформуючи те, як контент виглядає на веб-сторінці, і що ще на ньому доповнює цей вміст. Хоча часто використовується разом з HTML, він фактично не залежить від нього і може використовуватися з будь-якою мовою розмітки на основі XML.

JavaScript використовується в стеку веб-розробки. Отже, його використовувати можна як для фронтенду так і для бекенду. Розробники інтерфейсу часто використовують JavaScript, щоб зробити веб-сторінки динамічними. JavaScript є чудовим інструментом для таких завдань, як

перевірка форм подання або оновлення окремих частин вмісту сторінки без одночасного оновлення всієї сторінки. Без JavaScript Інтернет був би заповнений веб-сторінками, які просто відображають текст.

В моєму випадку я використовую Javascript тільки для придання анімацій та динамічності сторінки, оскільки це є одним з найзручніших інструментів для реалізації таких задач. Отже, підсумовуючи все вищезгадане у цьому підрозділі моя фронт-енд частина буде складатися з HTML, CSS та Javacript.

Для реалізації логіки програми, написання складних функцій, тобто створення бек-енд частини свого проекту я обрав мову програмування Python. Мій вибір зумовлений багатьма причинами та перевагами які надає Python у порівнянні з іншими мовами програмування. Основною перевагою Python є його простий синтаксис і довжина короткого коду. З великою кількістю документації Python дуже легко знайти інформацію по будь-якій функції, бібліотеці тощо. Python добре знайомий з універсальністю, і він добре розроблений. Програмне забезпечення, розроблене за допомогою Python (розробка бекенда за допомогою python), можна використовувати в різних операційних системах без необхідності в інтерпретаторі.

Зайве говорити, що Python полегшує завдання залучення до сучасних стратегій програмування. Python має кілька потужних бібліотек з величезною кількістю попередньо написаного коду. Тому розробникам не потрібно писати код з нуля, що прискорює час розробки. Це робить його ідеальним вибором для використання Python для розробки бекенда. Для мене існує три основних критерії, чому я вважаю прекрасно підходить для розробки мого додатку:

1. Використання об'єктно-орієнтованого програмування
2. Легко зрозуміти код
3. Фреймворк Django

Об’єктно-орієнтовані можливості є найважливішими для програмування. Python – це мова програмування, яка спеціально розроблена з різними об’єктно-орієнтованими поняттями. За допомогою об’єктно-

орієнтованого програмування різні способи поведінки та властивості організовуються в кілька об’єктів. Тому кожен має свою функцію. При об’єктно-орієнтованому програмуванні, якщо в якійсь частині коду виникає помилка, на решту коду це не впливає.

Основною перевагою розробки програмного забезпечення на Python є його читабельність. Основна причина його читабельності полягає у використанні пробілів для окреслення блоків коду. Очевидно, крапка з комою та другі дужки не використовуються в Python. Оскільки код легко читати, у розробників не буде проблем із читанням і розумінням коду, написаного іншим програмістом.

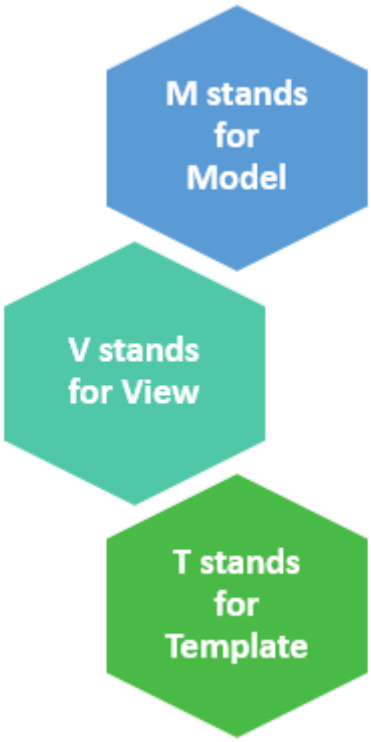
Ну й найголовнішим пунктом для мене, через який я обрав Python для бек-енду це фреймворк Django. Django використовується для розробки різних типів веб-сайтів, зокрема, програм, які можна налаштувати, наприклад, веб-сайту соціальних мереж, або як в моєму випадку для створення веб-сайту з планування часу. Django пропонує велику різноманітність функцій, які роблять процес розробки чистим та ефективним. Він поєднує в собі багато функцій, що робить його повною структурою. Ось список деяких основних функцій, які пропонує Django:

1. Open-Source
2. Швидка розробка
3. Масштабованість
4. Безпечність
5. Широко підтримувані бібліотеки
6. Інтерфейс адміністратора

Також, для мене важливо, що Python має вбудовану базу даних SQLLite, а Django забезпечує просте підключення до неї, з мінімальними затратами часу на це.

Django дотримується власної конвенції архітектури Model-View-Controller (MVC) під назвою Model View Template (MVT). MVT — це шаблон

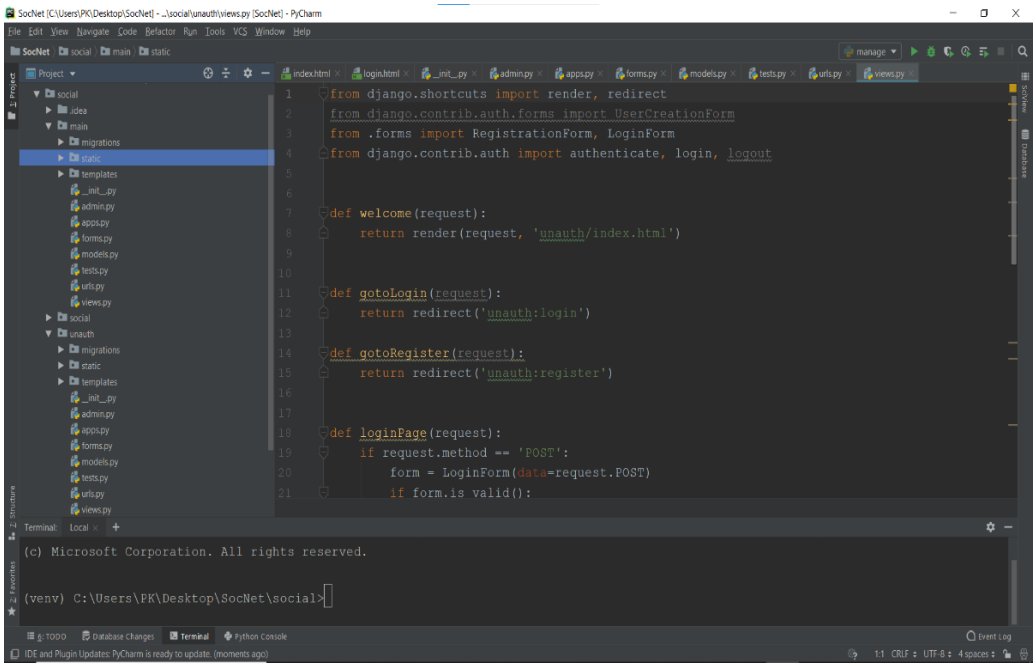
проектування програмного забезпечення, який в основному складається з 3 компонентів: Model , View та Template.



Використовуючи цей архітектурний шаблон для розробки додатку можна досягти зручного дизайну програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми.

2.2 Опис та пояснення вибору середовища розробки

PyCharm — одна з найпопулярніших IDE Python. Для цього існує безліч причин, включаючи той факт, що він розроблений JetBrains, розробником популярної IDE IntelliJ IDEA, яка є однією з трьох великих середовищ Java IDE та «найрозумнішої JavaScript IDE» WebStorm. Підтримка веб-розробки за допомогою Django є ще однією важливою причиною.



Основною причиною створення цієї IDE Pycharm було програмування на Python і робота на кількох платформах, таких як Windows, Linux та macOS. IDE містить інструменти аналізу коду, дебаггер, інструменти тестування, а також параметри контролю версій. Він також допомагає розробникам створювати плагіни Python за допомогою різних доступних API. IDE дозволяє нам працювати з кількома базами даних безпосередньо, не інтегруючи їх з іншими інструментами. Хоча він спеціально розроблений для Python, файли HTML, CSS і Javascript також можна створювати за допомогою цієї IDE. Він також має чудовий інтерфейс користувача, який можна налаштувати відповідно до потреб за допомогою плагінів.

Компанія JetBrains, яка є розробником цієї IDE, надає безкоштовний доступ усім студентам для користування версією “Professional”, яка надає

набагато більший функціонал, ніж безкоштовна версія “Community edition”, а саме:

1. Інтелектуальний редактор Python
2. Графічний дебаггер і тестовий запуск
3. Навігація та рефакторинг
4. Перевірки коду
5. Підтримка VCS
6. Наукові інструменти
7. Веб-розробка
8. Веб-фреймворки Python
9. Python Profiler
10. Можливості віддаленої розробки

Тому, щоб скористатися усіма перевагами, які надає компанія JetBrains, я буду використовувати PyCharm Professional 2022 року збірки.

РОЗДІЛ 3
ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1 Розробка методів та компонентів роботи інформаційної системи

В уніфікованій мові моделювання (UML) діаграма компонентів описує, як компоненти з’єднуються разом, щоб утворити більший компонент або програмну систему. Вони використовуються для ілюстрації структури як завгодно складних систем.

Діаграми компонентів дозволяють переконатися, що функціональність, необхідна системі, є прийнятною. Ці діаграми також використовуються як інструмент спілкування між розробниками та зацікавленими сторонами системи. Програмісти та розробники використовують діаграми для формалізації дорожніх карт впровадження, що дає їм змогу приймати кращі рішення щодо розподілу чи покращень. Системні адміністратори можуть використовувати діаграми компонентів для планування наперед, використовуючи уявлення про логічні компоненти програмного забезпечення та їх взаємозв’язки в системі.

Діаграма компонента розширює інформацію в елементі імені компонента. Одним із способів описати наданий і необхідний інтерфейс з цим компонентом є у вигляді прямокутного відсіку, прикріпленого до компонентного елемента. Іншим прийнятним способом представлення інтерфейсу є використання графічної конвенції «м'яч і гнізда». Дана залежність компонента від інтерфейсу представлена суцільною лінією компонента, що використовує інтерфейс, або від «льодяника», або від кульки, позначеної назвою інтерфейсу. Необхідна залежність використовуваного компонента від інтерфейсу ілюструється півколом або сокетом, позначеним ім’ям інтерфейсу, з’єднаним суцільною лінією з компонентом, якому потрібен цей інтерфейс. Традиційний інтерфейс можна відобразити за допомогою льодяника з символом каретки перед написом імені. Щоб проілюструвати їх

взаємозв’язок, з’єднайте розетку з льодяником суцільною лінією зі звичайними стрілками.

На малюнку 3.1 показана схема компонентів системи, що показує взаємодію між компонентами системи та формування системи в цілому.

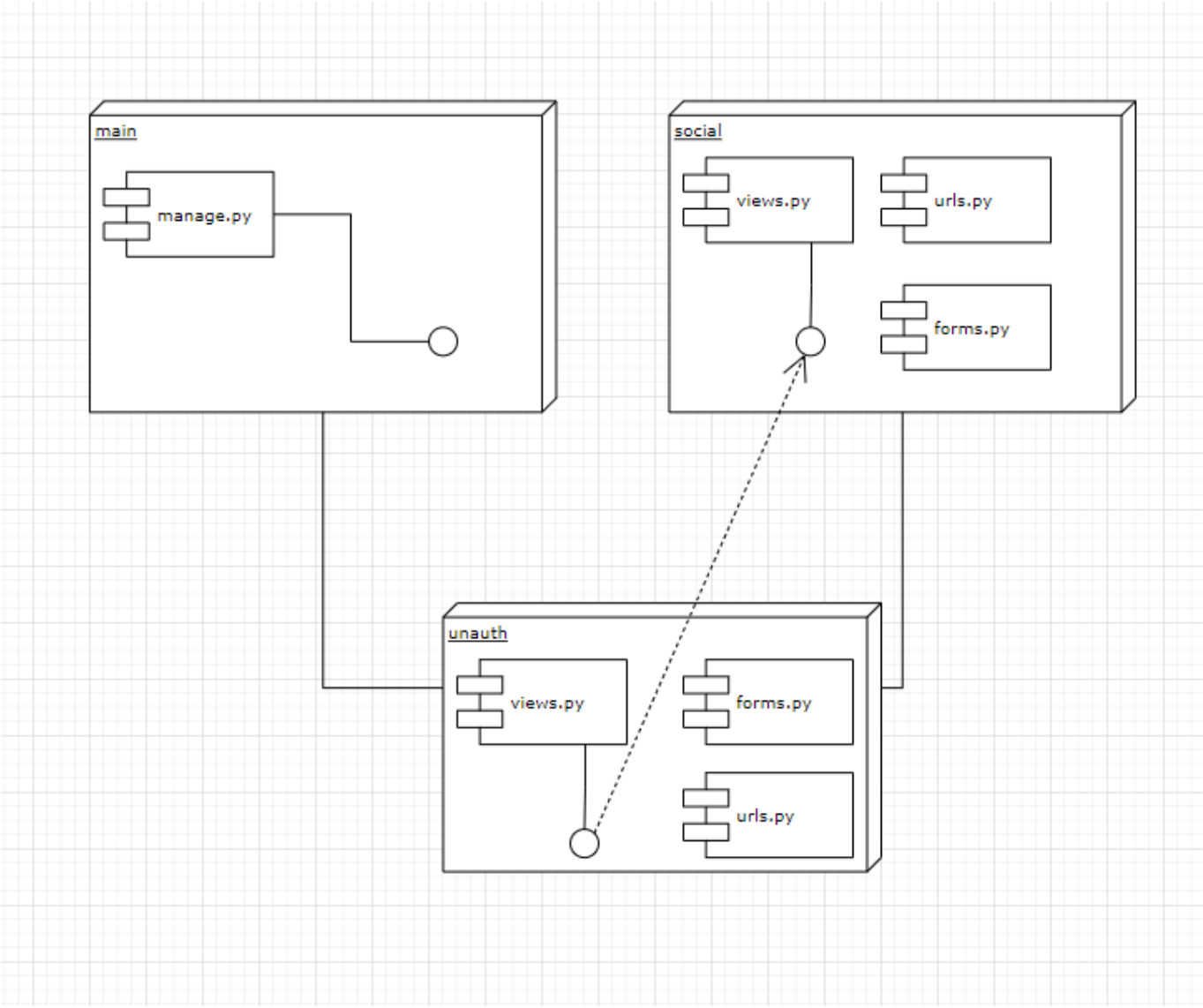


Рисунок 3.1 — Діаграма компонентів

3.2 Розробка алгоритмів та моделі

Основне функціональне призначення розроблюваної моделі полягає в обробці волонтерами заявок адміністраторів на обробки засмічених ділянок. Тобто, основна діяльність полягає в публікації і відгуках на заявки. Для проектування роботи моделі підтримки екологічної рівноваги використано UML-діаграму варіантів використання (рис. 3.2), яка описує діяльність користувачів в системі.

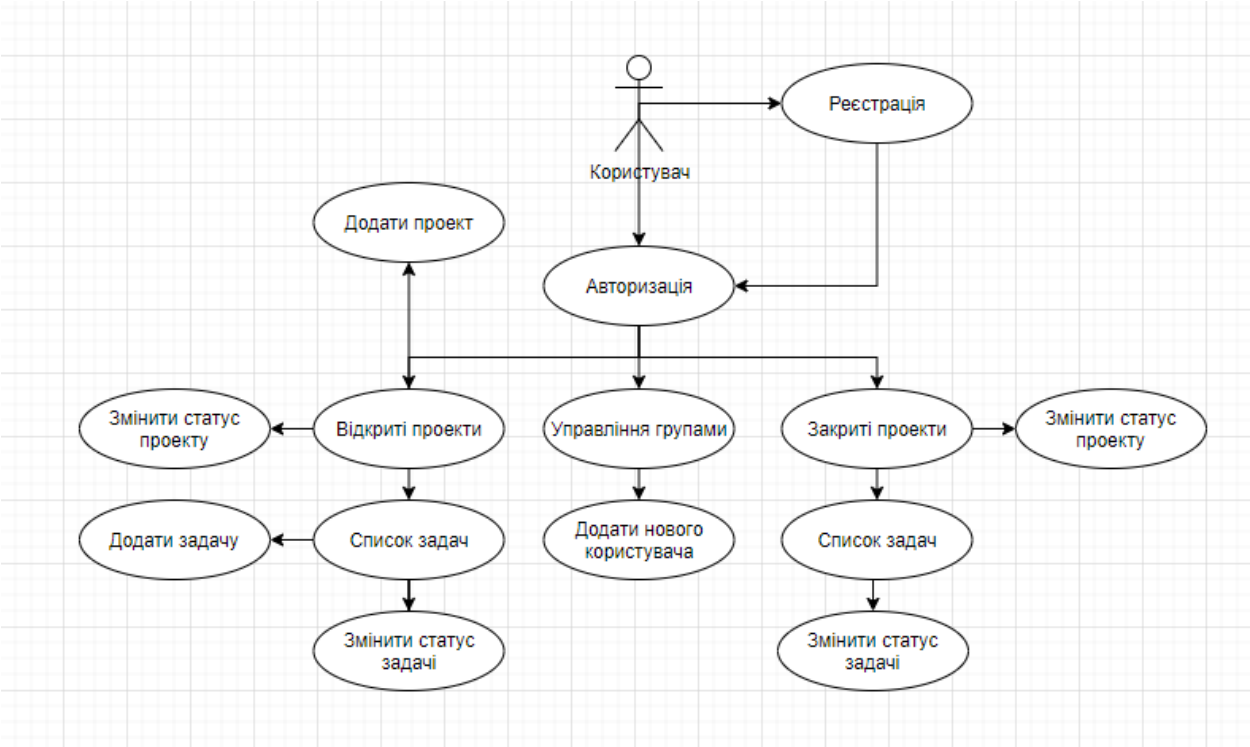


Рисунок 3.2 — Діаграма варіантів використання

3.3 Розробка внутрішньої структури

У програмній інженерії діаграма класів уніфікованої мови моделювання (UML) — це статична структурна діаграма, яка описує структуру системи, показуючи класи системи, їх властивості, операції (або методи) і зв’язки між об’єктами.

Діаграми класів є основним будівельним блоком об'єктно-орієнтованого моделювання. Використовується для загального концептуального моделювання структури програми та детального моделювання для перетворення моделей у програмний код. Діаграми класів також можна використовувати для моделювання даних. Класи на діаграмі класів представляють основні елементи програми, взаємодії та класи, що програмуються.

На рисунку 3.3 показані класи, що відображають структуру програмної реалізації розробленої інформаційної системи.

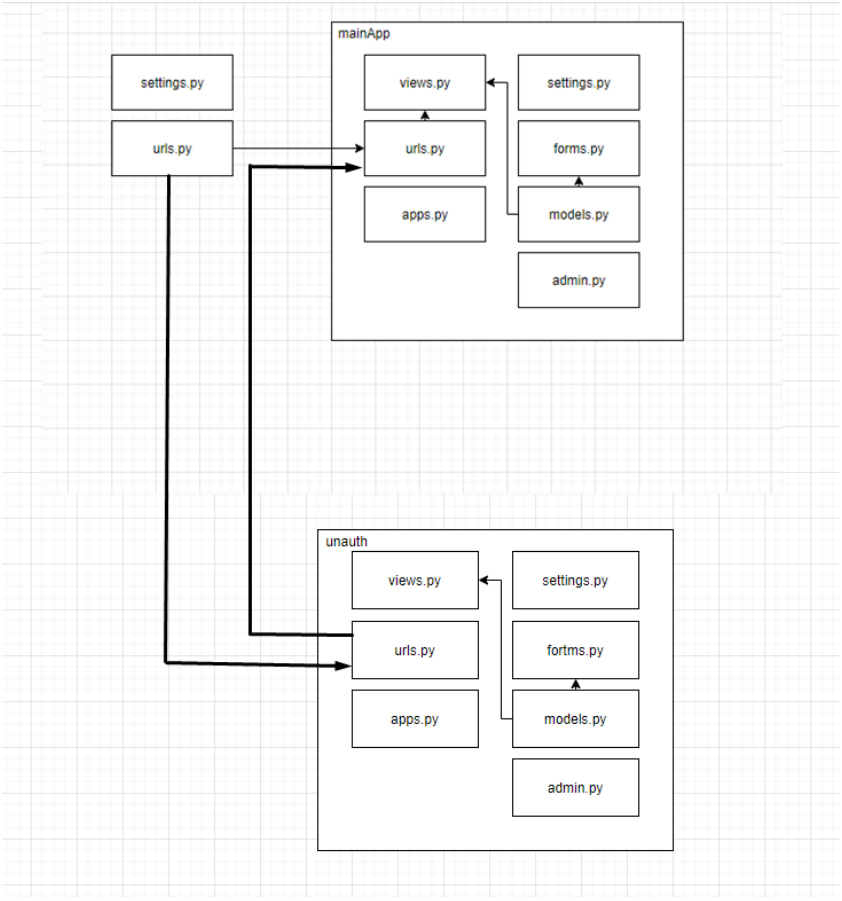


Рисунок 3.3 — Діаграма класів

3.4 Розробка графічного інтерфейсу користувача та опис логіки програми

Графічний інтерфейс складається з великої множини сторінок, проте серед них можна виділити список основних, а саме:

- сторінка проектів (рис. 3.5);
- сторінка проекту (рис. 3.6);
- сторінка створення проекту (рис. 3.7);
- сторінка задачі (рис. 3.8);
- сторінка створення задачі (рис. 3.9);
- сторінка редагування доступів до проекту (рис. 3.10);
- сторінка профілю (рис. 3.11).

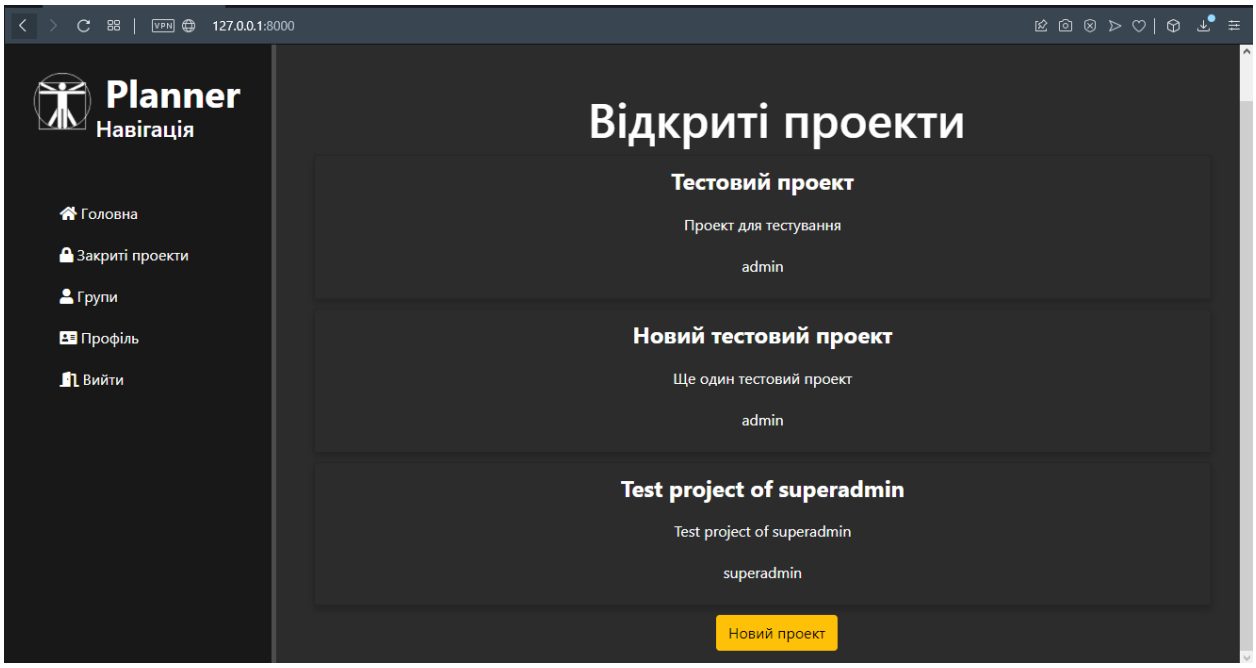


Рисунок 3.5 — Сторінка проектів

На рисунку 3.5 ми можемо бачити як виглядає вікно з проектами, до яких належить користувач(його було додано, або він створив його сам). Він може перейти вже в існуючий проект, або створити новий за потреби.

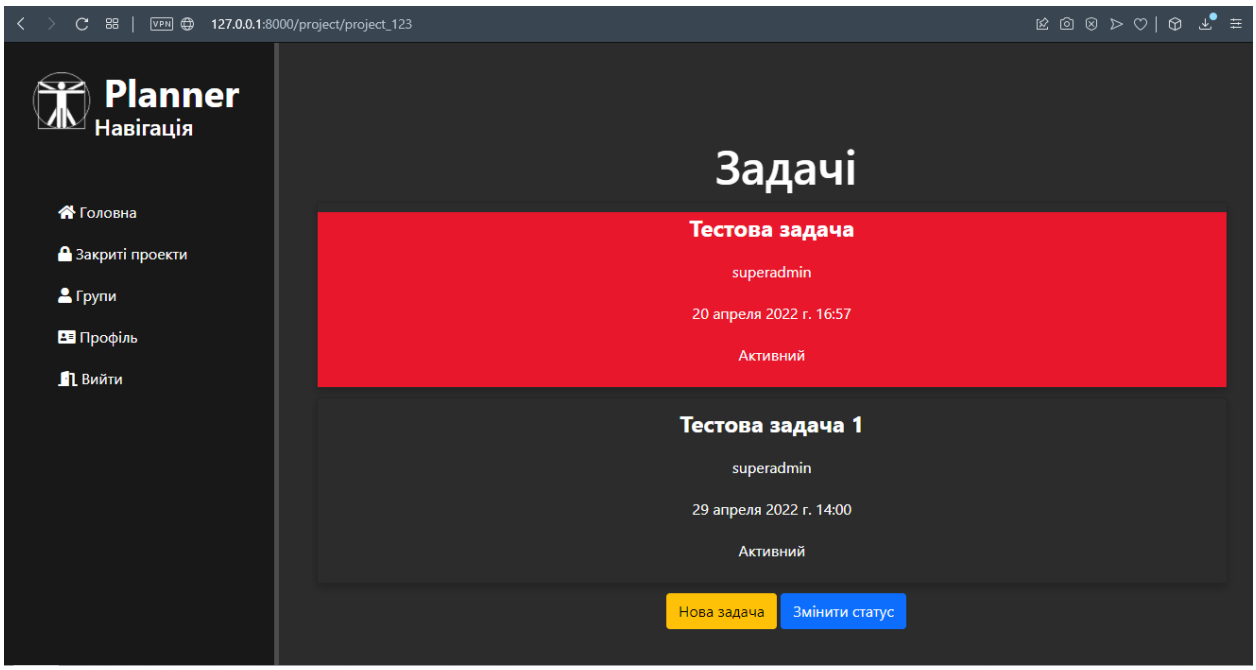


Рисунок 3.6 — Сторінка проекту

На рисунку 3.6 відображені задачі в проекті, червоним світяться ті, які прострочені, жовтим ті, яким залишилось менше години. Також, можна додати нову задачу, або змінити статус задачі на виконаний.

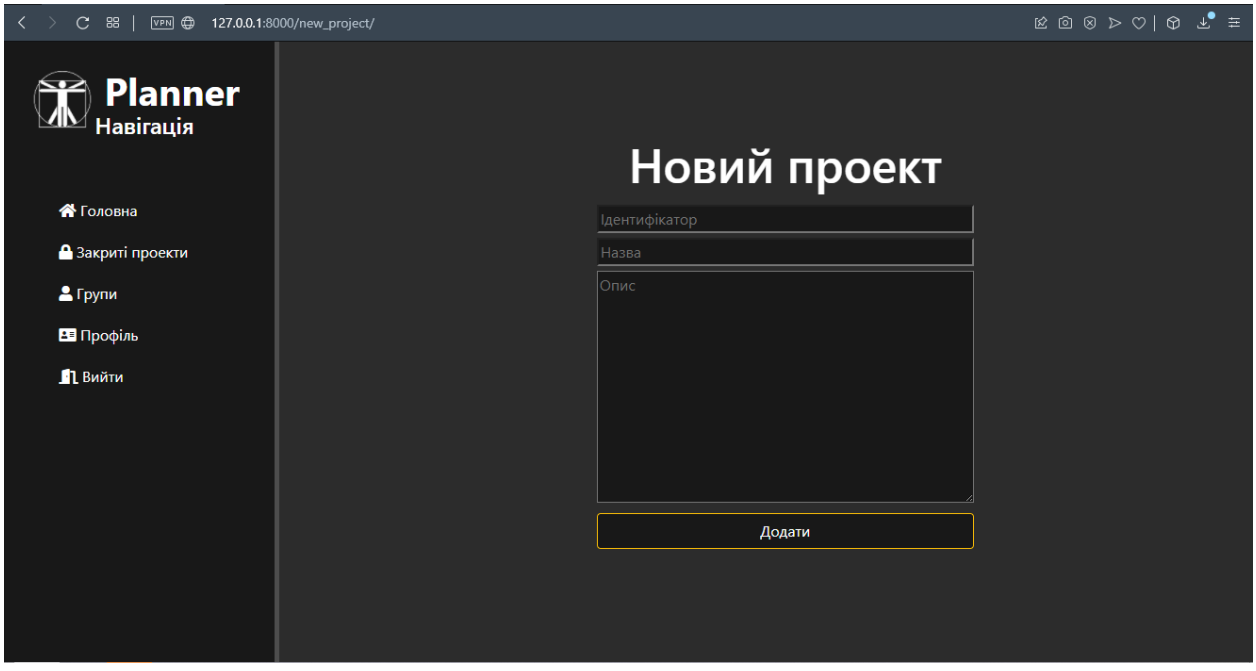


Рисунок 3.7 — Сторінка створення проекту

На рисунку 3.7 видно, як виглядає вікно створення проекту, потрібно ввести індентифікатор, назву та опис проекту.

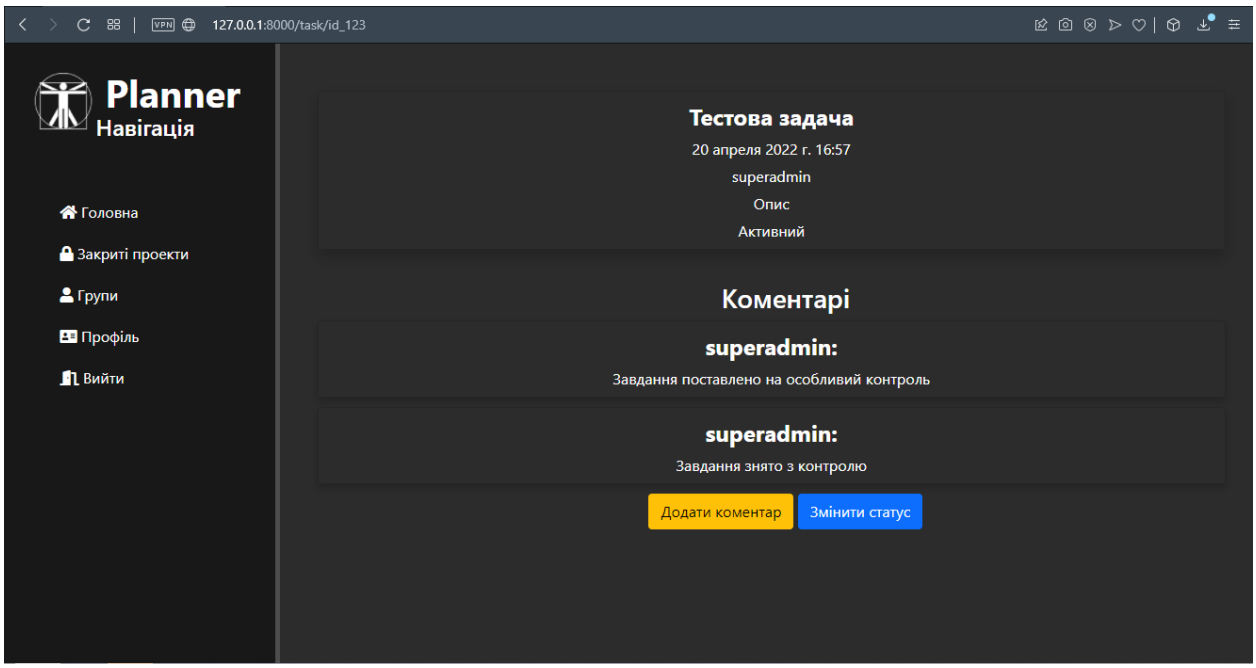


Рисунок 3.8 — Сторінка задачі

На рисунку 3.8 можна побачити як виглядає сторінка задачі, варто звернути увагу на кнопку «додати комантар», за домогою її було реалізовано спілкування з приводу якоїсь задачі між користувачами.

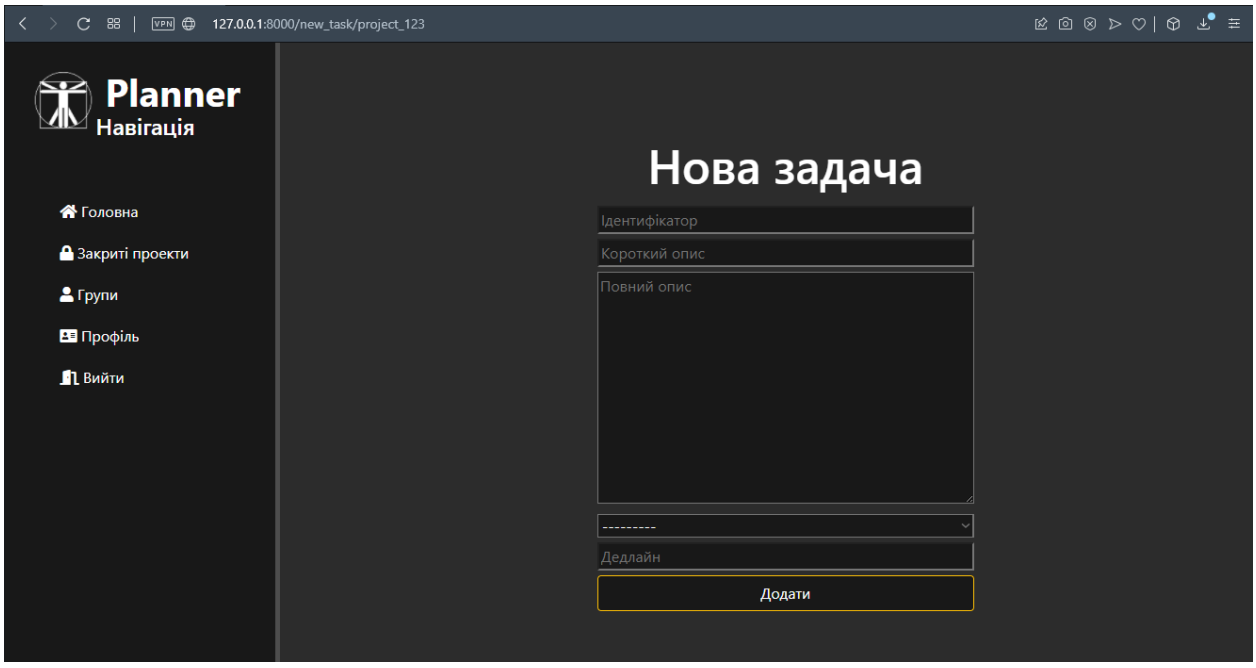


Рисунок 3.9 — Сторінка створення задачі

На рисунку 3.9 видно, як виглядає вікно створення задчі, потрібно ввести індентифікатор, назву та опис проекту та вказати дедлайн і задача з’явиться у пулі задач.

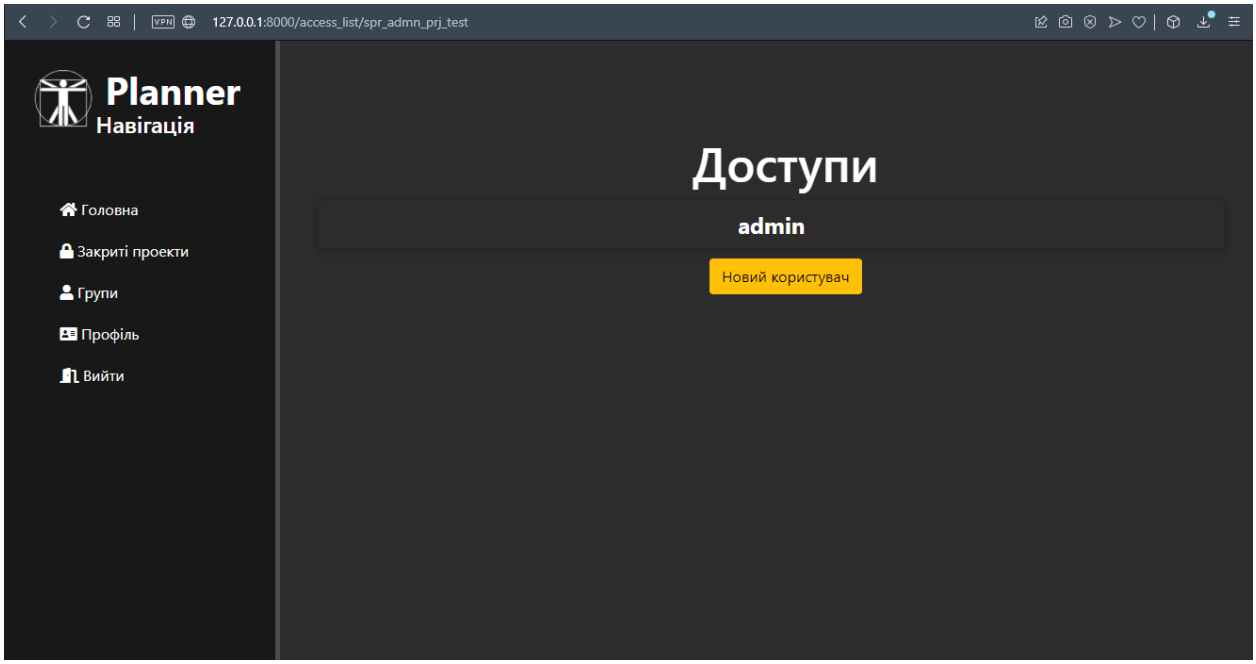


Рисунок 3.10 — Сторінка редагування доступів до проекту

На рисунку 3.10 зображено можливість для адміна надавати доступ іншим користувачам до якогось проекту, в якому керівником є адмін. Це зроблено для того, щоб прослідкувалась певна ієрархія між доступами різних користувачів.

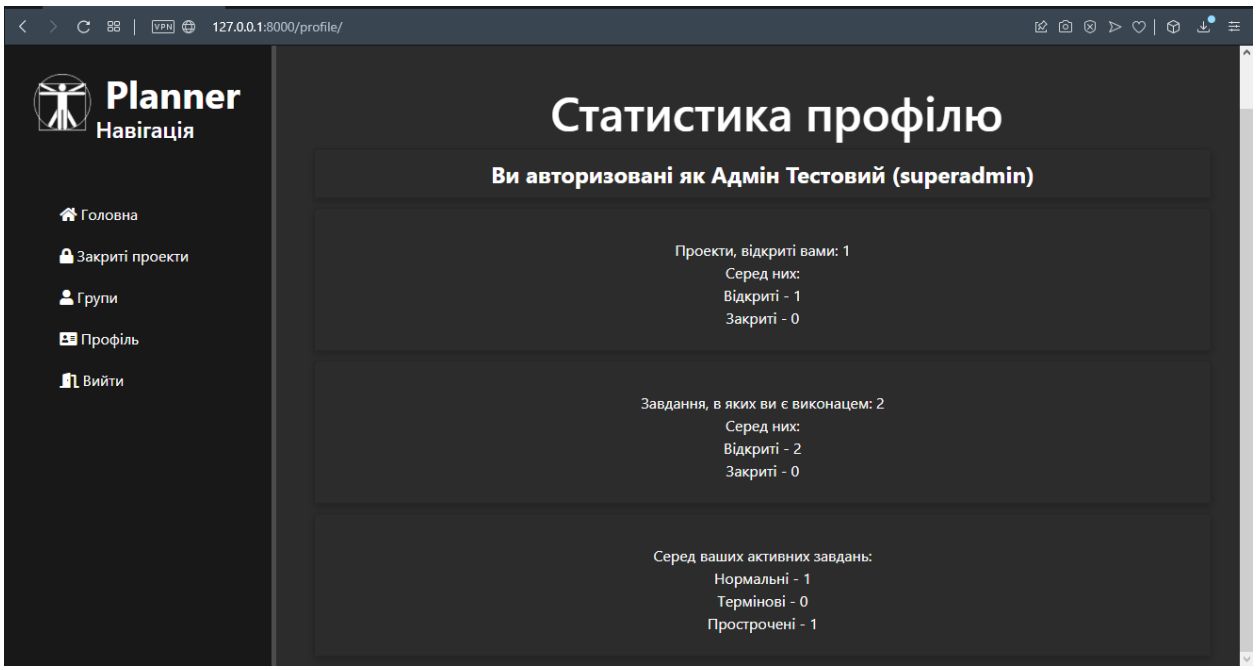


Рисунок 3.11 — Сторінка профілю

На рисунку 3.11 ми можемо побачити сторінку профіля кожного користувача, на якій мін може відслідковувати власну продуктивність.

ВИСНОВКИ

Основною метою роботи є створення власної реалізації веб-додатку для планування часу.

Для досягнення поставленої мети було виконано наступні завдання:

- провести аналіз поняття тайм-менеджменту;
- провести аналіз поняття веб-додатку;
- провести аналіз поняття веб-розробки;
- провести огляд мови програмування;
- провести огляд середовища розробки;
- розробити методи та компоненти системи;
- розробити алгоритми і моделі;
- розробити внутрішню структуру;
- розробити графічний інтерфейс користувача.

Завдяки чіткому виконанню завдань, поставлених на початку роботи, в результаті отримано повноцінний веб-додаток для планування часу, готовий до використання в реальних умовах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PyCharm [Електронний ресурс] – https://towardsdatascience.com/4-tips-to-get-the-best-out-of-pycharm-99dd5d01932d.
2. How to choose programming language [Електронний ресурс] – https://www.geeksforgeeks.org/how-to-choose-a-programming-language-for-a-project/
3. [Електронний ресурс] Django databases - https://docs.djangoproject.com/en/4.0/ref/databases/ [Електронний ресурс] Mobile phone in education - https://www.frontiersin.org/articles/10.3389/feduc.2020.00016/full
4. [Електронний ресурс] MVC Pattern - https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80
5. [Електронний ресурс] Як обрати мову програмування - https://itbukva.com/promo/16919-kak-vybrat-yazyk-programmirovaniya-dlya-proekta.html
6. [Електронний ресурс] What is Asana?- https://nira.com/asana-kanban/
7. [Електронний ресурс] How to work with Trello - https://startpack.ru/application/fog-trello
8. [Електронний ресурс] What is a web-app – https://helpx.adobe.com/ru/dreamweaver/using/web-applications.html
9. [Електронний ресурс] Що таке тайм-менеджмент? - https://kupibo.com.ua/dotrymannya-tajm-menedzhmentu/
10. [Електронний ресурс] 7 Trello analogs - https://weeek.net/ru/blog/trello-alternatives
11. [Електронний ресурс] FrontEnd Development - https://www.udemy.com/course/frontend-html-css-javascript/

12. [Електронний ресурс] Is JavaScript for FrontEnd or BackEnd? - https://careerkarma.com/blog/javascript-front-end-or-back-end/
13. [Електронний ресурс] Django Web Framework- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django