

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Модель інтелектуальної системи оптимізації проєктів ІТ компанії»

Студентки 2м курсу, 2м групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Сологуб
Лілії Вікторівни

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Савченко
Тетяна Віталіївна

Гарант освітньої програми
доктор економічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис гаранта

Токар Володимир
Володимирович

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«10» листопада 2020 р.

Завдання

на випускний кваліфікаційний проєкт студентіві

Сологуб Лілії Вікторівні

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Модель інтелектуальної
системи оптимізації проєктів ІТ компанії»

Затверджена наказом ректора від «30» листопада 2020 р. № 3224

2. Строк здачі студентом закінченого проєкту 25 листопада 2021р.

3. Цільова установка та вихідні дані до проєкту

Мета проєкту - дослідження процесу реалізації проєктів та організації
роботи над ними

Об'єкт дослідження - ІТ компанія

Предмет дослідження розробка моделі інтелектуальної системи оптимізації
проєктів ІТ компанії

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ МЕХАНІЗМІВ РЕАЛІЗАЦІЇ ПРОЄКТІВ В ІТ-КОМПАНІЯХ

1.1. Особливості ралізації проєктів в ІТ-компаніях

1.2. Інтелектуальні системи, та їх роль для оптимізації роботи над проєктами

1.3. Аналіз наявних програмних засобів для розподілу завдань при роботі над проєктами

1.4 Висновки до розділу 1

РОЗДІЛ 2. ВИБІР ЗАСОБІВ РОЗРОБКИ ТА ПРОЄКТУВАННЯ МОДЕЛІ РОЗПОДІЛУ ЗАВДАНЬ ПРИ РОБОТІ НАД ПРОЄКТАМИ

2.1. Методики управління проєктами

2.2. Функціональна модель системи розподілу завдань

2.3. Програмне забезпечення та технології розробки для реалізації моделі

2.4 Висновки до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОПТИМІЗАЦІЇ ПРОЄКТІВ

3.1. Загальна характеристика програмного забезпечення

3.2. Модуль роботи з завданнями

3.3 Модулі роботи з користувачами

3.4. Додаткові функції

3.5 Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

КЕРІВНИЦТВО КОРИСТУВАЧА

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проєкт магістра</i>	06.11.2020	06.11.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. Аналіз механізмів реалізації проєктів в іт-компаніях</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. Вибір засобів розробки та проєктування моделі розподілу завдань при роботі над проєктами</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. реалізація моделі інтелектуальної системи оптимізації проєктів</i>	21.06.2021	21.06.2021
8.	<i>Розділ 4. (Об'єднання розділів 3 та 4)</i>	20.09.2021	20.09.2021
9.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
10.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
11.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
12.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
13.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	03.11.2021	03.11.2021
14.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
15.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	22.11.2021 – 25.11.2021	22.11.2021
16.	<i>Здача зброшурованої випускного кваліфікаційного проєкту</i>	25.11.2021	25.11.2021
17.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	26.11.2021	26.11.2021
18.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «10» листопада 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту Савченко Т.В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Токар В.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Сологуб Л.В.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

У випускному кваліфікаційному проєкті проведено аналіз механізмів реалізації проєктів в ІТ – компаніях, проаналізовано роль програмних засобів для організації роботи над проєктами. Здійснено розробку функціональної моделі інформаційної системи для оптимізації проєктів та реалізацію самої моделі в вигляді веб-застосунку.

Ключові слова: ІТ-компанія, управління проєктами, ІТ-проєкти, веб-застосунок, таск-менеджер

ANNOTATION

The final qualification project analyzes the mechanisms of project implementation in IT companies, analyzes the role of software for organizing work on projects. A functional model of the information system for project optimization and implementation of the model in the form of a web application has been developed.

Keywords: IT company, IT projects, web application, task manager for projects, project management

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 АНАЛІЗ МЕХАНІЗМІВ РЕАЛІЗАЦІЇ ПРОЄКТІВ В ІТ-КОМПАНІЇ	5
1.1. Особливості реалізації проєктів в ІТ компаніях.....	5
1.2. Інтелектуальні системи, та їх роль для оптимізації роботи над проєктами.....	8
1.3 Аналіз наявних програмних засобів для розподілу завдань при роботі над проєктами.....	11
1.4. Висновки до розділу 1.....	16
РОЗДІЛ 2 ВИБІР ЗАСОБІВ РОЗРОБКИ ТА ПРОЄКТУВАННЯ МОДЕЛІ РОЗПОДІЛУ ЗАВДАНЬ ПРИ РОБОТІ НАД ПРОЄКТАМИ	17
2.1. Методики управління проєктами.....	17
2.2. Функціональна модель системи розподілу завдань.....	22
2.3. Програмне забезпечення та технології розробки для реалізації моделі.....	27
2.4. Висновки до розділу 2.....	33
РОЗДІЛ 3 РЕАЛІЗАЦІЯ МОДЕЛІ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОПТИМІЗАЦІЇ ПРОЄКТІВ	34
3.1. Загальна характеристика програмного забезпечення.....	34
3.2. Модуль роботи з завданнями.....	34
3.3. Модулі роботи з користувачами.....	42
3.4. Додаткові функції.....	44
3.5. Висновки до розділу 3.....	46
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ТЕХНІЧНЕ ЗАВДАННЯ	51
ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ	56
КЕРІВНИЦТВО КОРИСТУВАЧА	58
ДОДАТКИ	60

					<i>КНТЕУ 121 02м- 20.МР</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Модель інтелектуальної системи оптимізації проєктів ІТ компанії	<i>Стадія</i>	<i>Арк.</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		22.12.20		3	2	50
Керівник		Савченко Т.В.		22.12.20		Факультет інформаційних технологій 2м курс, 2м група		
Гарант		Токар В.В.		22.12.20				
Розробив		Сологуб Л.В.		22.12.20	<i>Зміст</i>			

ВСТУП

Актуальність. На сьогоднішній день багато компаній працюють над складними, дороговартісними проектами, робота над якими вимагає великих затрат часу та уваги з боку працівників. Декілька, навіть складних проектів, можуть вестися паралельно, що в свою чергу ще більше ускладнює розподіл задач між усіма учасниками проекту та контроль їх виконання. Так як всі компанії прагнуть збільшити прибутки шляхом успішного виконання більшої кількості проектів, вони зацікавлені в тому, щоб максимально ефективно використовувати свої ресурси.

Тому все більш актуальним стає використання засобів, які мають за мету допомогти в плануванні та контролюванні процесів, які виникають під час роботи над проектами.

Деякі проміжні завдання не можуть бути реалізовані, поки не завершені інші завдання; деякі завдання можуть здійснюватися тільки паралельно, і так далі. Якщо порушується синхронізація виконання різних завдань, весь проект може бути поставлений під загрозу. [1].

Розподіл обов'язків, встановлення дедлайнів, контроль над виконанням завдань є невід'ємними елементами в роботі над проектами будь-якої складності. Неправильно організовані дії на одному з цих етапів, можуть призвести до того, що проект буде зданий невчасно, або взагалі не буде виконаний належним чином. Навіть при високому рівні професіоналізму виконавців, без чітко організованої роботи, проект приречений на провал.

Організація, розподіл завдань, контроль їх виконання без програмних засобів займає дуже багато часу і сил, потребує значної концентрації уваги і розумових затрат керівника проекту, це викликає розумове перенавантаження та стрес, що в свою чергу може спричинити помилки в розподілі навантаження

Зм.	Арк.	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02м- 20.МР</i>			
Зав. каф.		Криворучко О.В.		27.02.21	Модель інтелектуальної системи оптимізації проектів ІТ компанії	Стадія	Арк.	Аркушів
Керівник		Савченко Т.В.		27.02.21		<i>В</i>	<i>3</i>	<i>50</i>
Гарант		Токар В.В.		27.02.21		<i>Факультет інформаційних технологій</i> <i>2м курс, 2м група</i>		
Розробив		Сологуб Л.В.		27.02.21				
					<i>Вступ</i>			

на працівників, випускання з виду важливих деталей, і навіть прискорення професійного вигорання.

Управління завданнями є важливою складовою управління проєктами, яка допомагає сегментувати проєкт на пов'язані та послідовні завдання. Кожне завдання планується, розподіляється та контролюється індивідуально. Тому для ефективної організації роботи над проєктами необхідно використовувати програмні засоби.

Програмне забезпечення для управління завданнями допомагає командам працювати злагоджено, використовуючи такі функції, як планування завдань, звітування про хід та відстеження часу, витраченого на виконання кожного завдання, а також допомагає дотримуватися термінів виконання своїх проєктів [2].

Мета дослідження: є дослідження процесу реалізації проєктів та організації роботи над ними.

Об'єкт дослідження: ІТ компанія.

Предмет дослідження: розробка моделі інтелектуальної системи оптимізації проєктів ІТ компанії.

Завдання дослідження:

- дослідити роль програмних засобів для організації роботи над проєктами в ІТ компанії.
- розробити модель інтелектуальної системи оптимізації проєктів ІТ компанії

Наукова новизна дослідження полягає в тому, що в процесі виконання проєкту буде створено повноцінний веб-застосунок, зі зручним і зрозумілим інтерфейсом, який при цьому матиме усі необхідні функції та автоматичний аналіз завантаженості працівників для оптимізації роботи над проєктами.

Методи дослідження: загальнонауковий аналітичний метод, порівняльний метод, системний метод, методи теорії бази даних, метод програмування.

					КНТЕУ 121 02м - 20.МР	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

РОЗДІЛ 1

АНАЛІЗ МЕХАНІЗМІВ РЕАЛІЗАЦІЇ ПРОЄКТІВ В ІТ-КОМПАНІЯХ

1.1. Особливості реалізації проєктів в ІТ компаніях

Проєкт – це комплекс дій, скерованих на одержання унікального результату, що може бути продуктом чи послугою.

Проєкти характеризуються певними спільними ознаками й властивостями, які виділяють їх в окремий вид діяльності:

- скеровані на досягнення конкретних цілей;
- містять у собі координоване виконання взаємозалежних дій;
- мають обмежену тривалість у часі, з певним початком і кінцем;
- певною мірою неповторні і унікальні.

Проєкти створюються в усіх сферах діяльності людини: в промисловості, будівництві, торгівлі, медицині, освіті, культурі, науці, охороні навколишнього середовища, соціальній сфері і т.д. І тому існує класифікація проєктів за областями застосування, за масштабом, складністю, термінами реалізації, характером та рівнем учасників [3].

Термін «ІТ-проєкт» використовується для позначення діяльності, пов'язаної з використанням або створенням деякої інформаційної технології. ІТ-проєкти передбачають розробку програмних додатків, створення інформаційних систем, розгортання ІТ – інфраструктури [1].

Особливості сучасних ІТ-проєктів:

- складність описання (достатньо велика кількість функцій, процесів, елементів даних і складні взаємозв'язки між ними), що потребує моделювання й аналізу даних і процесів;
- наявність сукупності підсистем, що тісно взаємодіють і мають свої

					<i>КНТЕУ 121 02м- 20.МР</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Модель інтелектуальної системи оптимізації проєктів ІТ компанії	<i>Стадія</i>	<i>Арк.</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		16.04.21		<i>РІ</i>	5	50
Керівник		Савченко Т.В.		16.04.21		Аналіз механізмів реалізації проєктів в ІТ компаніях	Факультет інформаційних технологій 2м курс, 2м група	
Гарант		Токар В.В.		16.04.21				
Розробив		Сологуб Л.В.		16.04.21				

локальні задачі та цілі функціонування;

- необхідність інтеграції існуючих і повторно розроблених застосувань;
- функціонування в неоднорідному середовищі на декількох апаратних платформах;
- роз'єднаність і різнорідність окремих груп розробників за рівнем кваліфікації і сформованих традицій використання тих або інших інструментальних засобів [4];

ІТ - проєкт – це комплекс дій, спрямованих на створення програмного забезпечення. Програмне забезпечення - це цифровий, а не фізичний продукт. Він складається з ідей, конструкцій, інструкцій та формул, а не з бетону та цегли. Програмний продукт – це сукупність програм і супровідної документації по їх установці, настройці, використанню і доопрацюванню. Згідно стандарту [5] життєвий цикл програми, програмної системи програмного продукту включає розробку, розгортання, підтримку і супровід. Якщо програмний продукт достатньо складний, то його розгортання у клієнтів, як правило, реалізується окремими самостійними проєктами впровадження. Супровід включає усунення критичних несправностей в системі і реалізується часто не як проєкт а, як процесна діяльність. Підтримка полягає в розробці нової функціональності, переробці вже існуючої функціональності, у зв'язку із зміною вимог, і поліпшенням продукту, а також усунення некритичних зауважень до програмного забезпечення виявлених при його експлуатації. Життєвий цикл програмного продукту завершується виведенням продукту з експлуатації і зняттям його з підтримки і супроводу.

Проєкти з розробки програмного забезпечення є складними з точки зору управління, тому що робота в даній галузі сама по собі ускладнена, оскільки вона базується на логічній, а не на фізичній роботі. Ця складність призводить до необхідності вищого спілкування та співпраці між членами команди

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		6

проекту. Розширене спілкування насправді є причиною того, що додавання людських ресурсів в ІТ-проекти не обов'язково вирішує проблему виконання термінів і насправді може погіршити ситуацію. Для успішної реалізації ІТ-проекту вкрай важливе налагоджене, конструктивне спілкування між членами команди, та підтримання документації в належному стані, що забезпечить передачу необхідної інформації, заощаджує час і сили [6].

Для ІТ – проектів характерний високий рівень невизначеності, тому що дуже складно точно визначити детальні вимоги до програмного продукту до початку проекту та важко оцінити складність проекту з розробки програмного забезпечення. Тому необхідно розроблювати план з управління ризиками, яких може бути досить багато, і передбачати певний резерв бюджету та часу.

Робота, яку виконують програмісти, нематеріальна – це колективні думки і ідеї, виражені на мові програмування. Через унікальність галузі досвід, накопичений в галузях матеріального виробництва, мало сприяє успіху в управлінні програмним проектом. Прямі аналогії з цими галузями не працюють. Управляти розробкою програмного забезпечення треба інакше. Не існує єдиного правильного процесу розробки. Ефективний виробничий процес повинен ґрунтуватися на ітеративності, інкрементальності, самокерованості команди і адаптивності. Головний принцип: не люди повинні будуватися під вибрану модель процесу, а модель процесу повинна підстроюватися під конкретну команду, щоб забезпечити її найвищу продуктивність.

Щоб програмний проект став успішним, необхідно:

- чітко ставити цілі;
- визначати спосіб досягнення мети;
- контролювати і управляти реалізацією;
- аналізувати погрози і протидіяти їм;
- створювати команду [7].

Отже, підсумовуючи все вище викладене можна сказати, що ІТ-проекти

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

багатогранні і динамічні, потребують гнучкого підходу до роботи над ними, чіткого встановлення цілей, розподілу завдань, контролю над виконанням завдань і планування часу й навантаження.

1.2 Інтелектуальні системи, та їх роль для оптимізації роботи над проєктами

Інтелектуальна інформаційна система (ІС) — це один з видів автоматизованих інформаційних систем, інколи ІС називають системою, засновану на знаннях. ІС є комплексом програмних, лінгвістичних і логіко-математичних засобів для реалізації основного завдання: здійснення підтримки діяльності людини і пошуку інформації в режимі розширеного діалогу на природній мові.

Класифікація задач, які вирішують інтелектуальні інформаційні системи:

- інтерпретація даних. Це одне з традиційних завдань для експертних систем. Під інтерпретацією розуміється процес визначення змісту даних, результати якого мають бути погодженими і коректними.
- діагностика. Під діагностикою розуміється процес співвідношення об'єкта з деяким класом об'єктів і виявлення несправності в деякій системі. Несправність – це відхилення від норми;
- моніторинг. Основне завдання моніторингу – безперервна інтерпретація даних у реальному масштабі часу і сигналізація про вихід тих або інших параметрів за допустимі межі;
- проєктування. Проєктування полягає в підготовці специфікацій на створення «об'єктів» із задалегідь визначеними властивостями. Під специфікацією розуміється весь набір необхідних документів - креслення, записка пояснення і так далі;
- прогнозування. Прогнозування дозволяє передбачати наслідки деяких подій або явищ на підставі аналізу наявних даних. Прогнозуючі

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

системи логічно виводять ймовірні наслідки із заданих ситуацій;

- планування. Під плануванням розуміється знаходження планів дій, що відносяться до об'єктів, здатних виконувати деякі функції;
- навчання. Під навчанням розуміється використання комп'ютера для навчання деякої дисципліни або предмету. Системи навчання діагностують помилки при вивченні якої-небудь дисципліни за допомогою електронної обчислювальної машини і підказують правильні рішення;
- керування. Під керуванням розуміється функція організованої системи, що підтримує певний режим діяльності. Таким чином здійснюється управління поведінкою складних систем відповідно до заданих специфікацій;
- підтримка прийняття рішень. Підтримка прийняття рішень — це сукупність процедур, що забезпечує особу, що приймає рішення, необхідною інформацією і рекомендаціями, що полегшують процес ухвалення рішення. Вони допомагають фахівцям вибрати і сформулювати потрібну альтернативу серед множини виборів при ухваленні відповідальних рішень [8];

Прикладом використання інтелектуальних систем в галузі проєктного менеджменту є програмні засоби для роботи над проєктами.

Проєкт - це комплекс скоординованих дій на одержання унікального результату, що може бути продуктом чи послугою, обмежений за часом, якістю та термінами. Проєкти складні вже за самою своєю суттю. Вони включають в себе виконання чисельних взаємозалежних дій. В окремих випадках ці взаємозв'язки досить очевидні, в інших випадках вони мають більш тонку природу [1].

Деякі проміжні завдання не можуть бути реалізовані, поки не завершені інші завдання; деякі завдання можуть здійснюватися тільки паралельно, і так

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		9

далі. Якщо порушується синхронізація виконання різних завдань, весь проєкт може бути поставлений під загрозу. Проєкт - це система, тобто ціле, що складається із взаємозалежних частин, причому система динамічна, і, отже, потребує особливих підходів до управління.

Робота над проєктами - дуже кропіткий процес, особливо над великими і складними. Ключем до успішного і своєчасного виконання будь-якого проєкту є грамотно організована робота над ним.

Розподіл обов'язків, встановлення дедлайнів, контроль над виконанням завдань є невід'ємними елементами в роботі над проєктами будь-якої складності. Неправильно організовані дії на одному з цих етапів, можуть призвести до того, що проєкт буде зданий невчасно, або взагалі не буде виконаний належним чином. Навіть при високому рівні професіоналізму виконавців, без чітко організованої роботи, проєкт приречений на провал.

Організація, розподіл завдань, контроль їх виконання без програмних засобів займає дуже багато часу і сил, потребує значної концентрації уваги і розумових затрат керівника проєкту, це викликає розумове перенавантаження та стрес, що в свою чергу може спричинити помилки в розподілі навантаження на працівників, випускання з виду важливих деталей, і навіть прискорення професійного вигорання.

Управління завданнями є важливою складовою управління проєктами, яка допомагає сегментувати проєкт на пов'язані та послідовні завдання. Кожне завдання планується, розподіляється та контролюється індивідуально. Тому для ефективної організації роботи над проєктами необхідно використовувати програмні засоби.

Програмне забезпечення для управління завданнями допомагає командам працювати злагоджено, використовуючи такі функції, як планування завдань, звітування про хід та відстеження часу, витраченого на виконання кожного завдання, а також допомагає дотримуватися термінів виконання своїх проєктів.

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Додатковими функціями системи управління завданнями є: планування бюджету, синхронізація електронної пошти, списки справ та нагадування про завдання.

Отже, використання програмного забезпечення для роботи над проектами є справді актуальним, оскільки вони прискорюють виконання завдань, допомагають в плануванні та контролюванні процесів та оптимізують використання ресурсів компаній.

1.3. Аналіз наявних програмних засобів для розподілу завдань при роботі над проектами

На даний момент існує велика кількість різноманітного програмного забезпечення для роботи над проектами. Це можуть бути як великі програми, для виконання цілого комплексу завдань, так і вузько направлені, які виконують лише певні функції, безкоштовні і платні. Найбільшою популярністю користуються програми для управління завданнями - таск-менеджери.

Таск-менеджер - це програмне рішення, призначене для управління завданнями при роботі над проектами. Його функціональність дозволяє керівнику роздавати персоналізовані та групові завдання, розділити їх на етапи, контролювати виконання всіх заданих процесів. Таск-менеджери досить давно користуються популярністю, бо вони дозволяють здійснювати загальне управління та контроль над проектами з меншими затратами часу [9].

Типові функції таск-менеджерів:

- сповіщення: відстеження ключових етапів та повідомлення учасників процесу про нові завдання, оновлення, помилки та досягнення;
- редагування: редагування та оновлення завдань на основі потреби проекту або наявності ресурсів;
- відстеження прогресу завдання: відстеження ходу виконання

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

завдання протягом його життєвого циклу та контроль ключових показників ефективності;

- планування завдань: складання графіку та розподіл завдань окремим особам чи командам, встановлення дати початку та кінцевого терміну під час створення завдання;
- позначення завдань: створення тегів та міток для завдань та підзавдань, щоб упорядкувати та класифікувати їх, додавати контекст та збирати інформацію, пов'язану із завданнями;
- створення підзавдань: розділення завдання на менші підзавдання для управління завданнями на макрорівні;
- управління повторюваними завданнями: створення завдань та подій, які повторюються в певний час та день протягом тижня, місяця чи року;
- відстеження часу: відстеження колективного часу, витраченого всіма учасниками процесу на поставлені завдання [2].

Згідно з дослідженнями американської компанії Capterra, яка є провідним безкоштовним ресурсом Інтернету для довідки про ділове програмне забезпечення, п'ятьма найпопулярнішими таск менеджерами є Asana, Basecamp, Jira, Todoist та Trello.

Trello – це таск-менеджер, заснований на принципі канбан-дошки. Канбан - це японська система розпорядку, що була розроблена в Toyota. На дошці всі справи поділені за станом виконання (наприклад, у три стовпчики: «зробити», «у процесі», «зроблено»). У Trello можна створити власну дошку з відповідною структурою, а потім кріпити на неї картки із завданнями, додавати до них дедлайн, мітку, файли. Працювати можна просто онлайн у браузері або завантаживши додаток.

Переваги:

- дуже простий інструмент, і його легко інтегрувати в робочий процес;
- базовий функціонал абсолютно безкоштовний, але широкий. Платні

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

- Asana Premium коштує 10,99\$ доларів за користувача на місяць, а Business – 24,99\$ доларів.

Todoist більше нагадує класичний список справ. Їх можна поєднувати в проекти, відмічаючи кольоровими мітками. Todoist дозволяє працювати в команді: проєкт можна зробити загальним, а кожне завдання закріпити за окремим виконавцем. Є можливість встановлювати терміни виконання й нагадування.

Переваги:

- інтерфейс Todoist простий і зрозумілий;
- завдання можна форматовувати: встановлювати напівжирний шрифт або курсив, додавати посилання;
- Todoist дозволяє додавати файли з комп'ютера або хмарних сховищ, а також голосові замітки;
- пошук за завданнями можна полегшити за допомогою фільтрів;
- Todoist підтримує інтеграцію з Gmail, Outlook, iCalendar, Google Календар та іншими популярними сервісами;
- завантажити Todoist можна майже на все: смартфони на Android та iOS, комп'ютери на Windows XP / 7 / 8, Windows 10, macOS, Linux і навіть на Apple Watch і Wear OS, а також доступні розширення для Google Chrome і Safari.

Недоліки:

- відсутня україномовна версія;
- синхронізація, нагадування, фільтри, мітки та інші важливі функції доступні лише у тарифі Преміум. Він коштує 3\$ долари за користувача на місяць [10].

Jira пропонує можливість управління проєктами кожному співробітнику: вони можуть налаштовувати, планувати завдання, а уповноважені фахівці контролювати їх роботу. Є web-версія, а також додатки для iOS і Android.

						<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
							14
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			

Платформа включає в себе канбан-дошки, змішані методології і Scrum, завдяки чому може охопити широке коло користувачів. У таск менеджері діє система оцінок. Всі проєкти і завдання можна розподілити за пріоритетністю. Представлені інструменти роблять роботу команди прозорою: кожен буде в курсі всього, що відбувається в колективі і наскільки ефективно він справляється зі своїми обов'язками. Доступна інтеграція зі сторонніми програмами [11].

Переваги:

- канбан-дошки, змішані методології і Scrum;
- система оцінок;
- пріоритетність завдань;
- інтеграція зі сторонніми сервісами.

Недоліки:

- дуже багато функцій, тому для деяких користувачів інтерфейс може виявитися складним;
- вартість користування, 10\$ - 14\$ на місяць за одного користувача.

Basecamр - таск-менеджер на основі чек-листів. Об'єднує в собі завдання, календар, групові чати, повідомлення про події та роботу з файлами. Універсальна програма, яка може замінити кілька додатків для роботи з завданнями, повністю організувати і автоматизувати роботу [12].

Переваги:

- прозорість інтерфейсу; всі проєкти на одному екрані, у кожного є підпис, коли він останній раз оновлювався;
- можна створювати необмежену кількість проєктів при будь-якому тарифі;
- безкоштовна версія для студентів

Недоліки:

- немає україномовної версії;
- ціна – 99\$ доларів за місяць користування.

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
Зм.	Арк.	№ докum.	Підпис	Дата		15

1.4. Висновки до розділу 1

Проект – це комплекс дій, скерованих на одержання унікального результату, що може бути продуктом чи послугою. Термін «ІТ-проект» використовується для позначення діяльності, пов'язаної з використанням або створенням деякої інформаційної технології. ІТ-проекти передбачають розробку програмних додатків, створення інформаційних систем, розгортання ІТ – інфраструктури. Проекти з розробки програмного забезпечення є складними з точки зору управління, тому що робота в даній галузі сама по собі ускладнена, оскільки вона базується на логічній, а не на фізичній роботі.

Робота над проектами - дуже кропіткий процес, особливо над великими і складними. Ключем до успішного і своєчасного виконання будь-якого проекту є грамотно організована робота над ним. Управління завданнями є важливою складовою управління проектами, яка допомагає сегментувати проект на пов'язані та послідовні завдання. Програмне забезпечення для управління завданнями допомагає командам працювати злагоджено а також допомагає дотримуватися термінів виконання своїх проектів.

На даний момент існує велика кількість різноманітного програмного забезпечення для роботи над проектами. Це можуть бути як великі програми, для виконання цілого комплексу завдань, так і вузько направлені, які виконують лише певні функції, безкоштовні і платні. Найбільшою популярністю користуються програми для управління завданнями - таск-менеджери.

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

РОЗДІЛ 2

ВИБІР ЗАСОБІВ РОЗРОБКИ ТА ПРОЄКТУВАННЯ МОДЕЛІ РОЗПОДІЛУ ЗАВДАНЬ ПРИ РОБОТІ НАД ПРОЄКТАМИ

2.1. Методики управління проєктами

На сучасному етапі створюється велика кількість проєктів в різних галузях. Проєкти дуже різноманітні. І наразі відсутня єдина ідеальна система управління проєктами, що підходила б для всіх видів проєктів і для кожного керівника, але є напрацьовані ефективні методи управління проєктами. Серед них виділяють наступні:

- класичний проєктний менеджмент Waterfall;
- Agile;
- Scrum;
- Lean;
- Kanban;
- Six Sigma;
- PRINCE2.

Класичний проєктний менеджмент Waterfall. Це методологія ведення проєкту, заснована на каскадному циклі, при якому етапи робіт йдуть послідовно і наступний етап починається тільки після завершення попереднього, відповідно і завдання передаються послідовно. Як правило, виділяють 5 етапів класичного проєктного менеджменту. Даний метод орієнтований на проєкти, які вимагають чіткої послідовності виконання завдань. Waterfall строго прив'язаний до часу виконання завдань, що заздалегідь визначається на етапі планування, і тому для реалізації проєктів в рамках

Зм.	Арк.	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02м- 20.МР</i>			
Зав. каф.		Криворучко О.В.		24.05.21	Модель інтелектуальної системи оптимізації проєктів ІТ компанії	Стадія	Арк.	Аркуші
Керівник		Савченко Т.В.		24.05.21		P2	17	50
Гарант		Токар В.В.		24.05.21		Факультет інформаційних технологій 2м курс, 2м група		
Розробив		Сологуб Л.В.		24.05.21				
					Вибір засобів розробки та проєктування моделі розподілу завдань при роботі над проєктами			

даного методу використовуються інструменти календарно-сітьового планування, зокрема діаграма Ганта.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню. Завдання, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями. Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу [13].

Перевагою класичного проєктного менеджменту є те, що він передбачає чіткі вимоги від замовника і керівництва компанії вже на першому етапі проєкту, чітке планування з правильним розподілом часу, врахуванням можливих ризиків і ускладнень, що забезпечує певну стабільність в роботі та дозволяє уникнути стресів.

Agile - методологія, орієнтована на розробку програмного забезпечення, з'явилася як відповідь на низьку ефективність методу Waterfall для управління складними проєктами. Основа цього методу – швидкий та гнучкий підхід до управління проєктами і продуктами, орієнтований на динамічне формування вимог і забезпечення їх реалізації в результаті постійної взаємодії всередині самоорганізованих робочих груп, що складаються з фахівців різного профілю. Відповідно до даного підходу, проєкт не розбивається на послідовні фази, він розбивається на маленькі підпроєкти, які потім «збираються» в готовий продукт. Ініціація і планування проводяться для всього проєкту, а наступні етапи: розробка, тестування і інші проводяться для кожного міні-проєкту окремо. Це дозволяє передавати результати цих міні-проєктів швидше, а приступаючи до нового підпроєкту (ітерації) в нього можна додати зміни без великих витрат і впливу на інші частини проєкту. Перевагою Agile є його

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		18

гнучкість і адаптивність, здатність підлаштовуватися під будь-які умови і процеси організації. Існує безліч методів, які базуються на ідеях Agile, найпопулярніші з яких - Scrum і Kanban [14].

Scrum розроблений для проєктів, в яких необхідні «швидкі перемоги» в поєднанні з толерантністю до змін. Відповідно до цього методу над проєктом працює невелика (5-9 чоловік) універсальна команда фахівців, члени якої повинні володіти більш ніж однією компетенцією, необхідною для реалізації проєкту, вміти працювати в команді, здійснювати взаємодопомогу та взаємозаміну. Крім фахівців до складу команди входить ще двоє людей: власник продукту та scrum-майстер. Власник продукту – це представник замовника, що стежить за розвитком проєкту. Scrum-майстер стежить, щоб ніхто не заважав команді самостійно і комфортно працювати над поставленими завданнями. Проєкт розбивається на частини, які відразу можуть бути використані замовником для отримання цінності, послідовність реалізації яких визначається їх пріоритетністю. Найважливіші «частини» першими відбираються для виконання. Робочий процес ділиться на рівні спринти. Зазвичай це періоди від тижня до місяця, в залежності від проєкту та команди. В кінці спринту замовнику видається робочий інкремент продукту – найважливіші «частини», які вже можна використовувати. Наприклад, сайт з частиною функціоналу або програма, яка вже працює, нехай і частково. Після цього команда проєкту приступає до наступного спринту. Основна структура процесів Scrum: упорядкування беклогу, планування спринту, щоденна літучка, підведення підсумків спринту і ретроспективи спринту.

У Lean, так само, як і в Scrum, робота розбивається на невеликі пакети поставки, які реалізуються окремо і незалежно. Але в Lean для розробки кожного пакету поставок існує потік операцій (workflow) з етапами. Як і в класичному проєктному менеджменті, це можуть бути етапи планування, розробки, виробництва, тестування і поставки – або будь-які інші необхідні для

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		19

якісної реалізації проєктів етапи.

Етапи Lean і їх гнучкість дозволяють бути впевненими в тому, що кожна частина проєкту реалізується так, як потрібно. У Lean не прописані чіткі межі етапів, як в Scrum прописані обмеження спринтів. Крім того, на відміну від класичного проєктного менеджменту, Lean дозволяє паралельно виконувати кілька завдань на різних етапах, що підвищує гнучкість і збільшує швидкість виконання проєктів [13].

Kanban. Згідно методу Kanban, інкремент продукту передається вперед з етапу на етап, а в кінці виходить готовий до постачання елемент. В Kanban дозволяється залишити незакінчене завдання на одному з етапів, якщо його пріоритет змінився і є інші термінові завдання. Наприклад, невідредагована стаття для блогу, підвішена без дати публікації, або частина коду функції, яку можливо не включатимуть в продукт.

Kanban не обмежує тривалість спринтів, дозволяє члену команди вести кілька завдань одночасно, не регламентує зустрічі за статусом проєкту – можна робити це як зручно, а можна не робити взагалі. Для роботи з Kanban необхідно визначити етапи потоку операцій (workflow). Вони зображуються як стовпці, а завдання позначають спеціальні картки. Картка переміщується по етапах і на кожному етапі відсоток завершення стає вище. На виході отримується готовий до постачання замовнику елемент продукту.

Головний показник ефективності в kanban - це середній час проходження завдання по дошці. Завдання пройшло швидко - команда працювала продуктивно і злагоджено. Завдання затягнулася - треба думати, на якому етапі та чому виникли затримки та чию роботу треба оптимізувати [14]. Kanban добре підходить для роботи досить згуртованих, мотивованих і досвідчених команд над проєктами, що не обмежуються жорсткими дедлайнами.

Six Sigma – методологія що використовується у корпоративному менеджменті для вдосконалення виробництва та усунення дефектів, в рамках якої проводяться заходи зі знаходження і виключення причин помилок або

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		20

дефектів у бізнес-процесах, шляхом зосередження на тих вихідних параметрах, які є критично важливими для споживача. Дана методологія дуже схожа на Kanban, тільки з встановленими етапами реалізації завдань - плануванням, визначенням цілей і тестуванням якості, Але вона передбачає більшу кількість зустрічей команди, більш структурований процес реалізації проєктів, ретельний вимір і контроль показників проєкту на етапах реалізації, що надає можливість постійно удосконалювати процеси реалізації проєкту.

Six Sigma підходить для важких проєктів, в яких багато нових і складних операцій. Даний підхід дозволяє реалізовувати елементи проєкту, вчитися на помилках і підвищувати якість в майбутньому.

PRINCE2 - британська методологія. Назва походить від акроніму «PRojects IN Controlled Environments version 2», що перекладається як «Проєкти в контрольованому середовищі версія 2». На відміну від гнучких методів, PRINCE2 не використовує ітеративний підхід до проєкту. Його можна порівняти з гібридом класичного підходу до проєктного управління та концентрації на якості з Six Sigma.

PRINCE2 концентрується на управлінських сторонах проєкту, виражених в 7 принципах, 7 процесах і 7 темах проєкту. 7 принципів визначають загальні правила управління проєктами по PRINCE2, визначають базу методології; 7 процесів визначають кроки просування по проєктному циклу; 7 тем – аспекти, за якими проводиться контроль для досягнення успіху проєкту. Крім того, PRINCE2 рекомендує адаптувати методологію під кожен конкретну організацію.

На початку проєкту PRINCE2 пропонує визначити 3 основних аспекти проєкту:

- бізнес-аспект (Чи принесе цей проєкт вигоду?);
- споживчий аспект (Який потрібен продукт, що ми будемо робити?);
- ресурсний аспект (достатньо у нас всього, щоб досягти мети?).

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		21

У PRINCE2 більш чітко визначена структура команди проєкту, ніж у більшості підходів до проєктного управління. Це пов'язано з тим, що PRINCE2 орієнтований на масштабні державні проєкти і великі організації. Згідно PRINCE2 у кожного члена команди є своя чітка роль в кожному з 7 процесів.

PRINCE2 може бути адаптований для проєктів будь-якого масштабу і будь-якої предметної області. Методологія пропонує конкретні рекомендації щодо зміни життєвого циклу проєкту, рольової моделі і набору обов'язкових документів відповідно до потреб проєкту [13].

Управління проєктами - це не точна наука, а область, в якій немає непорушних основ і універсальних рішень. Для різних проєктів необхідні різні методи та інструменти. Зважаючи на те, що найважливіше для проєкту - дедлайни, ресурси, дотримання процесу, або відразу кілька факторів - вибирається метод управління проєктом, орієнтований на досягнення цього показника.

Менеджеру проєкту дуже рідко вдається знайти метод, який ідеально підходить для проєкту. Здебільшого йому доводиться прикладати зусилля для створення і налаштування власних систем управління проєктами, складених з елементів існуючих систем або створених абсолютно з нуля.

2.2. Функціональна модель системи розподілу завдань

Найпоширенішим представником моделі системи розподілу завдань є таск-менеджер. Таск-менеджер – це спеціальне програмне забезпечення, яке допомагає формулювати та фіксувати завдання, контролювати хід роботи та закінчувати проєкти вчасно. Є додатки, які спрямовані на індивідуальну продуктивність людини, але більшість таск-менеджерів спрямовані на команди: спільна робота над проєктом, розподіл завдань і віддалений моніторинг ходу роботи. Цей інструмент допомагає грамотно організувати спільну роботу, витратити менше часу на зустрічі для обговорення та пояснення особисто

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		22

кожному його нових обов'язків. Саме таск-менеджери ідеальні для оптимізації та прискорення робочих процесів.

В даному проєкті, як модель інтелектуальної системи оптимізації проєктів в ІТ-компанії, було розроблено таск-менеджер. Дана програма забезпечує можливість призначати, коментувати та керувати завданнями, переглядати список користувачів та інформацію про них, додавати або видаляти користувачів, читати новини, додані іншими користувачами, або додавати свої власні.

Для проєктування описаного таск-менеджера були використані наступні UML діаграми, які створені за допомогою інструменту StarUML:

- діаграма варіантів використання;
- діаграма послідовності;
- діаграма діяльності;

Діаграма варіантів використання (Рис. 2.1.) дає найзагальніше представлення функціонального призначення системи. В даній системі є два типи користувачів – звичайний користувач і адміністратор. Звичайний користувач – це користувач, який пройшов реєстрацію в системі, і для початку роботи з нею повинен авторизуватися.

Після успішної авторизації він отримує доступ до:

- призначення і перегляду призначеного завдання на виконавця;
- редагування призначених завдань;
- видалення призначених завдань;
- перегляд завдань, які призначені на даного користувача;
- завершення завдань, які призначені на даного користувача;
- коментування завдань, які пов'язані з даним користувачем;
- перегляду списку користувачів, та інформації про них;
- стрічки новин даної компанії, де він може переглядати та створювати новини.

					КНТЕУ 121 02м - 20.МР	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

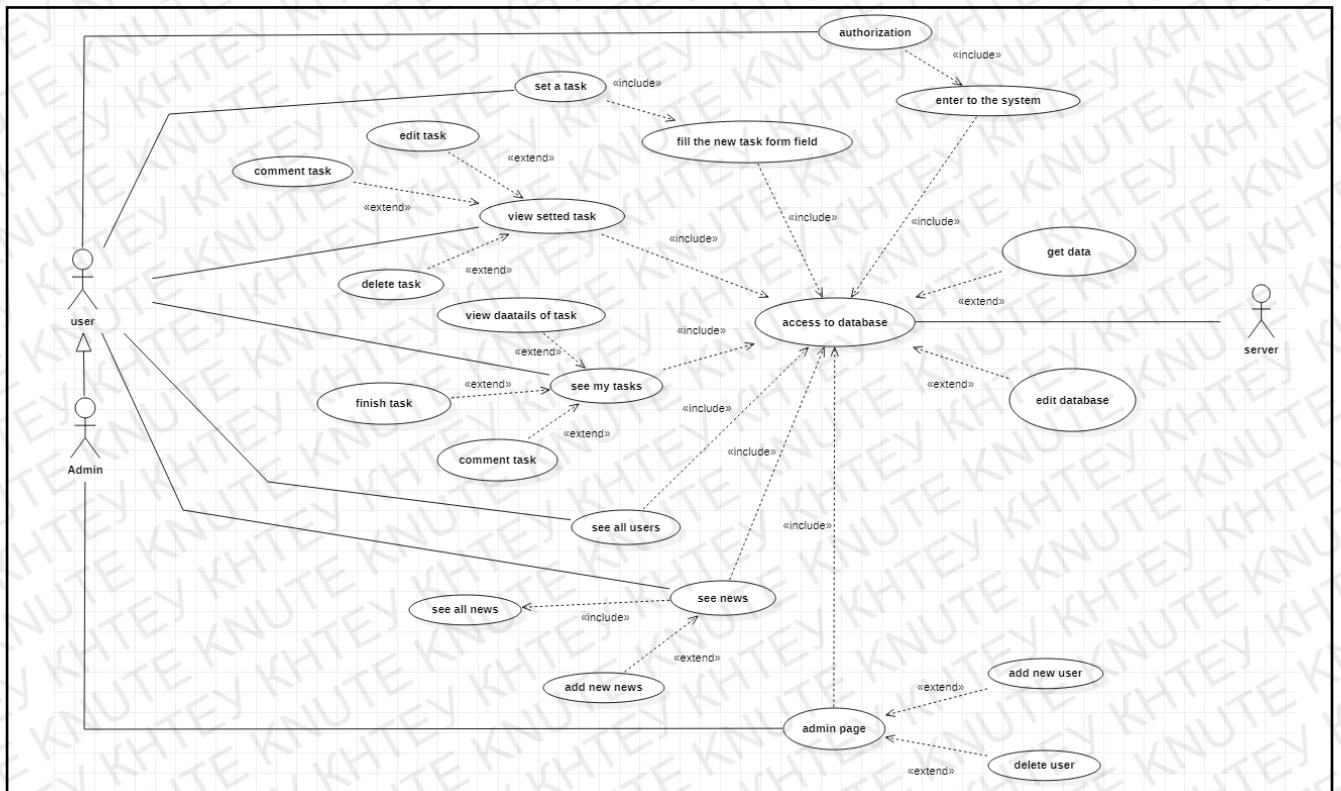


Рис. 2.1. Діаграма варіантів використання

Адміністратор отримує доступ до тих самих функцій, що й звичайний користувач, але з можливістю додавати нових або видаляти наявних користувачів.

Діаграма діяльності (Рис. 2.2.) зображує алгоритм дії системи. На початковому етапі алгоритму відбувається авторизація користувача в системі, шляхом введення логіну та паролю. Після цього він отримує доступ до таких функцій: робота з поставленими завданнями (поставити нове завдання або керувати вже поставленими завданнями), робота з «моїми завданнями», доступ до стрічки новин (перегляд або додавання новин), якщо користувач є адміністратором, то додатково може додавати або видаляти користувачів.

Якщо обирається варіант роботи з поставленими завданнями, то з'являється можливість або поставити нове завдання або керувати вже поставленими. В першому випадку користувач заповнює форму для нового завдання і в разі коректного заповнення, завдання призначається на виконавця і відображається в розділі «Поставлені завдання».

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		24

додавати власні новини, шляхом заповнення відповідної форми і завершити роботу в даному розділі.

У випадку, коли користувач є адміністратором, окрім усіх функцій простого користувача, він має доступ до сторінки адміністратора. На сторінці адміністратора він може видаляти наявних користувачів або додавати нових користувачів, шляхом заповнення відповідної форми і після цього завершити роботу в даному розділі. На цьому робота з системою завершується.

Діаграма послідовності (Рис 2.3). відображає порядок і моменти часу взаємодії об'єктів системи. Вона є записом протоколу конкретного сеансу роботи системи.

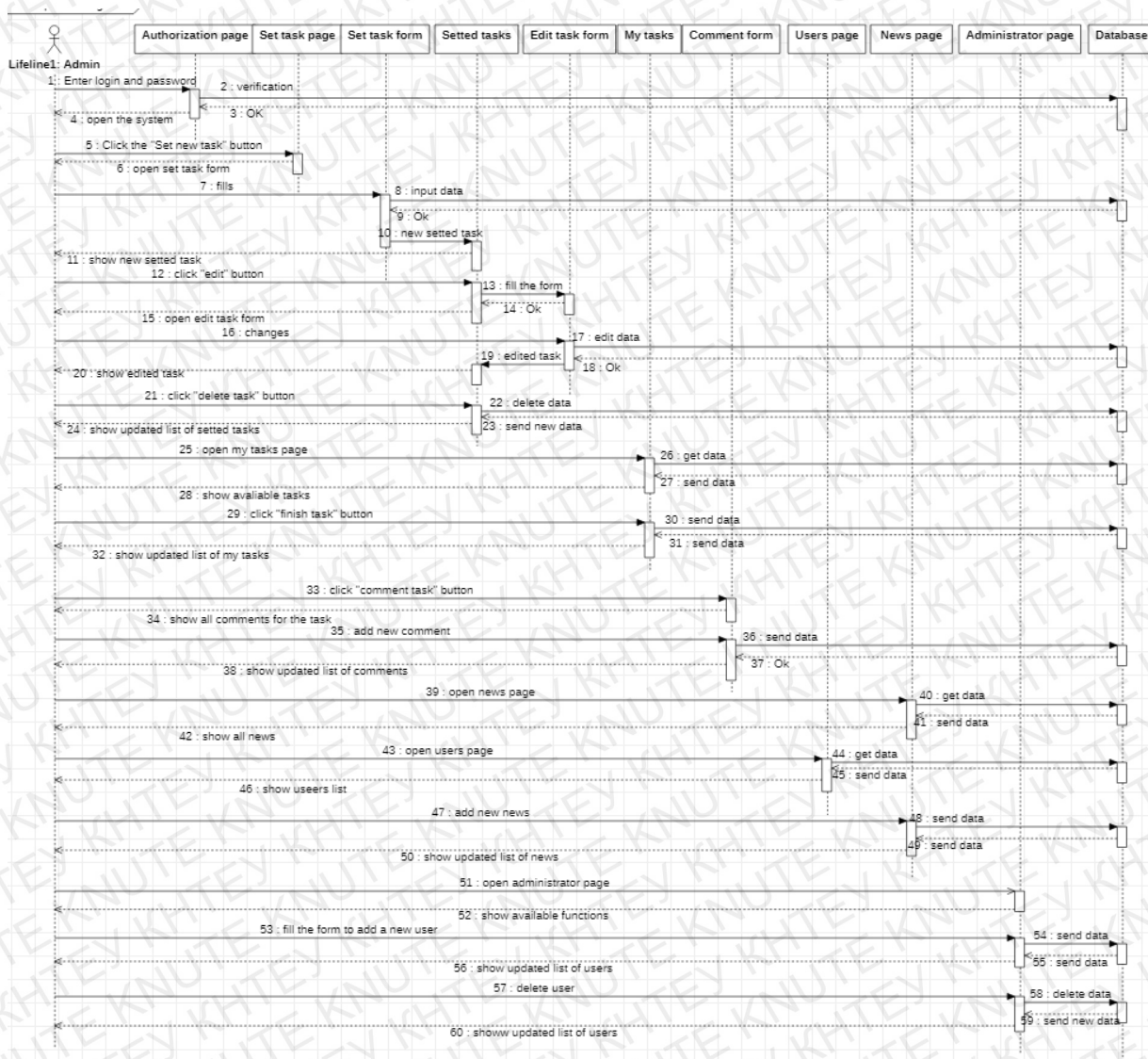


Рис 2.3. Діаграма послідовності

2.3. Програмне забезпечення та технології розробки для реалізації моделі

З огляду на тенденції розвитку технологій у світі, з ширшим використанням Інтернету, компанії змінюють спосіб роботи та надають перевагу онлайн сервісам. Перехід в онлайн дозволяє отримати цілий ряд переваг для роботи, тому більшість компаній в якості програмних рішень обирають саме веб-застосунки.

Веб-застосунок – це комп'ютерна програма, яка використовує веб-браузер для виконання певної функції, інтерфейсом якої є веб-сайт. Веб-браузер в цьому випадку використовується, як клієнт, а веб-сервер в якості сервера. У порівнянні з локальним програмним забезпеченням, доступ до веб-застосунків можна отримати з будь-якого комп'ютера через Інтернет [15].

Більшість веб-застосунків є кросплатформними, зазвичай мінімальною вимогою є веб-браузер. Це дає користувачеві можливість вибору того, коли і де він використовуватиме програму, сприяючи гнучкості роботи всередині компанії, для підвищення загальної продуктивності співробітників.

Веб-застосунок має багато переваг, зокрема:

- доступ до програми без встановлення локально на персональний комп'ютер;
- нижчі технічні вимоги до комп'ютера користувача;
- усі користувачі мають доступ до однієї версії, тому це усуває будь-які проблеми з сумісністю;
- можливість роботи веб-застосунку на декількох платформах, незалежно від операційної системи чи пристрою;
- доступ до програми будь-де за допомогою веб-браузера.

Веб-застосунки пропонують хороший спосіб безпечного доступу до централізованих даних. До серверів звертається безпосередньо особа або команда, які ними керують. Завдяки використанню хмарних обчислень сервери

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		27

можуть бути повністю резервними та реплікованими, щоб запобігти простою в результаті катастрофи. Це дозволяє уникнути необхідності підтримувати безпеку кожного пристрою, на якому використовується програма. Загалом ризик несанкціонованого доступу зменшується, а введення заходів безпеки стає простішим, оскільки це робиться централізовано [16].

Відповідно до цього даний таск-менеджер було розроблено у вигляді веб-застосунку з використанням технології, які використовуються в веб-розробці.

JavaScript - це текстова мова програмування, яка використовується як на стороні клієнта, так і на стороні сервера, що дозволяє зробити веб-сторінки інтерактивними. Якщо HTML і CSS є мовами, які надають структуру та стиль веб-сторінкам, JavaScript надає веб-сторінкам інтерактивні елементи, які взаємодіють з користувачем.

Використання JavaScript покращує користувацький досвід веб-сторінки, перетворюючи її зі статичної сторінки в інтерактивну, додає поведінку веб-сторінкам.

JavaScript в основному використовується для веб-додатків і веб-браузерів. Він також використовується за межами Інтернету в програмному забезпеченні, серверах та вбудованих апаратних елементах керування. Ось деякі основні речі, для яких використовується JavaScript:

- додавання інтерактивної поведінки до веб-сторінок. JavaScript дозволяє користувачам взаємодіяти з веб-сторінками. Таким чином роблячи їх живими та динамічними: змінювати колір сторінки та масштаб зображення, відтворювати аудіо, тощо;
- створення веб- та мобільних додатків. Розробники можуть використовувати різні фреймворки JavaScript для розробки та створення веб- та мобільних додатків. Фреймворки JavaScript — це колекції бібліотек коду JavaScript, які надають розробникам попередньо

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		28

написаний код для використання для рутинних функцій і завдань програмування — буквально фреймворк для створення веб-сайтів або веб-додатків. Популярні інтерфейсні фреймворки JavaScript включають React, React Native, Angular і Vue;

- створення веб-серверів і розробка серверних додатків. Крім веб-сайтів і програм, розробники також можуть використовувати JavaScript для створення простих веб-серверів і розвитку внутрішньої інфраструктури за допомогою Node.js [17].

Мова розмітки гіпертексту (HTML) - це мова комп'ютера, яка складається з більшості веб-сторінок та онлайн-додатків. Гіпертекст - це текст, який використовується для посилань на інші фрагменти тексту, тоді як мова розмітки - це серія маркування, яка повідомляє веб-серверам стиль і структуру документа.

HTML не вважається мовою програмування, оскільки він не може створювати динамічні функції. Натомість за допомогою HTML веб-користувачі можуть створювати та структурувати розділи, абзаци та посилання за допомогою елементів, тегів та атрибутів.

Найпоширеніші варіанти використання HTML:

- веб-розробка. Розробники використовують HTML-код, щоб розробити, як браузер відображає елементи веб-сторінки, такі як текст, гіперпосилання та медіафайли;

- Інтернет навігація. Користувачі можуть легко переміщатися та вставляти посилання між пов'язаними сторінками та веб-сайтами, оскільки HTML активно використовується для вбудовування гіперпосилань;

- веб-документація. HTML дозволяє впорядковувати та формувати документи, подібно до Microsoft Word;

Консорціум World Wide Web Consortium (W3C) підтримує та розробляє специфікації HTML, а також забезпечує регулярні оновлення [18].

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		29

Каскадні таблиці стилів (CSS) - це мова таблиць стилів, яка використовується для опису подання документа, написаного мовою розмітки, як-от HTML [19].

CSS розроблено для розділення відображення та вмісту, включаючи макет, кольори та шрифти [20]. Такий поділ може покращити доступність вмісту, забезпечити більшу гнучкість і контроль у специфікації характеристик відображення, дозволити кільком веб-сторінкам спільно використовувати форматування, вказавши відповідний CSS в окремому файлі .css. Розділення форматування та вмісту також дає можливість представити одну й ту саму сторінку розмітки в різних стилях для різних методів візуалізації, наприклад, на екрані, у друкованому вигляді, голосом (через мовленнєвий браузер або програму зчитування з екрана), а також на основі шрифту Брайля [21].

VueJS — це прогресивний JavaScript-фреймворк з відкритим кодом, який використовується для розробки інтерактивних веб-інтерфейсів. Це один із відомих фреймворків, що використовуються для спрощення веб-розробки. VueJS фокусується на шарі перегляду. Його можна легко інтегрувати у великі проекти для розробки інтерфейсу без будь-яких проблем.

Vue використовує синтаксис шаблону на основі HTML, який дозволяє прив'язувати відтворений DOM до даних базового екземпляра Vue. Усі шаблони Vue є дійсним HTML, який можна проаналізувати браузерами, що відповідають специфікації, і синтаксичними аналізаторами HTML. Vue компілює шаблони у віртуальні функції візуалізації DOM. Віртуальна модель об'єкта документа (або «DOM») дозволяє Vue відображати компоненти у своїй пам'яті перед оновленням браузера. У поєднанні з системою реактивності Vue може розрахувати мінімальну кількість компонентів для повторного візуалізації та застосувати мінімальну кількість маніпуляцій DOM при зміні стану програми.

Vue має систему реактивності, яка використовує прості об'єкти JavaScript

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		30

і оптимізоване повторне відтворення. Кожен компонент відстежує свої реактивні залежності під час візуалізації, тому система точно знає, коли потрібно повторно відобразити, і які компоненти повторно відобразити [22].

MongoDB - це документно-орієнтована база даних NoSQL, яка використовується для зберігання даних великого обсягу. NoSQL використовується як альтернатива традиційним реляційним базам даних. Бази даних NoSQL досить корисні для роботи з великими наборами розподілених даних. MongoDB - це інструмент, який може керувати документально-орієнтованою інформацією, зберігати або отримувати інформацію.

MongoDB підтримує різні форми даних. Це одна з багатьох технологій нереляційних баз даних, які виникли в середині 2000-х під прапором NoSQL - зазвичай для використання в програмах для великих даних та інших роботах з обробки даних, які погано вписуються в жорстку реляційну модель. Замість використання таблиць і рядків, як у реляційних базах даних, архітектура MongoDB складається з колекцій і документів.

Організації можуть використовувати MongoDB для спеціальних запитів, індексування, балансування навантаження, агрегації, виконання JavaScript на стороні сервера та інших функцій.

MongoDB використовує записи, які складаються з документів, які містять структуру даних, що складається з пар полів і значень. Документи є основною одиницею даних у MongoDB. Документи подібні до нотації об'єктів JavaScript, але використовують варіант під назвою Binary JSON (BSON). Перевага використання BSON полягає в тому, що він вміщує більше типів даних. Поля в цих документах подібні до стовпців у реляційній базі даних. Вміщувані значення можуть бути різних типів даних, включаючи інші документи, масиви та масиви документів, відповідно до посібника користувача MongoDB. Документи також включатимуть первинний ключ як унікальний ідентифікатор.

Набори документів називаються колекціями, які функціонують як

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		31

еквівалент таблиць реляційної бази даних. Колекції можуть містити будь-які типи даних, але обмеження полягає в тому, що дані в колекції не можуть бути розподілені між різними базами даних [23].

Node.js є середовищем виконання JavaScript з відкритим вихідним кодом. Хоча початковий випуск підтримував лише Linux, він розвинувся як справді міжплатформне середовище виконання JavaScript для розробки широкого спектру додатків. Середовище часу виконання інтерпретує JavaScript за допомогою двигуна JavaScript V8 від Google.

V8 - це механізм виконання з відкритим кодом, створений Google у 2008 році. Написаний на C++, V8 компілює вихідний код JavaScript у рідний машинний код замість того, щоб інтерпретувати його в реальному часі.

Основна функціональність Node.js реалізована в бібліотеці JavaScript. Прив'язки Node.js написані на C++. Node.js дозволяє створювати веб-сервери та мережеві інструменти за допомогою JavaScript та набору модулів, які обробляють різні основні функції. Надаються модулі для введення-виведення файлової системи, мереж (наприклад, HTTP, DNS, TCP, TLS/SSL або UDP), двійкових даних (буферів), функцій криптографії, потоків даних та інших основних функцій. Модулі Node.js використовують API, розроблений для зменшення складності написання серверних додатків

Node.js в основному використовується для створення мережевих додатків, використовуючи зв'язок у реальному часі між кінцевими точками. Node.js добре працює в програмах, керованих операцією введення-виведення.

Функції в Node.js розроблені так, щоб не блокувати роботу додатка. Команди виконуються паралельно, і використовують зворотний виклик, щоб повідомити про завершення або провал завдань.

Node.js надає веб-серверам програмування на основі подій, що дозволяє розробляти швидкі веб-сервери на JavaScript. Розробники можуть створювати високомасштабовані сервери без використання потоків, використовуючи

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		32

спрощену модель програмування на основі подій, яка використовує зворотні виклики для сигналу про завершення завдання [24]. Node.js був створений тому, що паралельність є складною в багатьох мовах програмування на стороні сервера і часто призводить до низької продуктивності. Node.js поєднує простоту мови сценаріїв з потужністю мережевого програмування.

2.4. Висновки до розділу 2

Отже, на сучасному етапі створюється велика кількість проектів в різних галузях. Проекти дуже різноманітні. І наразі відсутня єдина ідеальна система управління проектами, що підходила б для всіх видів проектів і для кожного керівника, але є напрацьовані ефективні методи управління проектами: Waterfall, Agile, Scrum, Lean, Kanban, Six Sigma, PRINCE2. Для різних проектів необхідні різні методи та інструменти. Зважаючи на те, що найважливіше для проекту - дедлайни, ресурси, дотримання процесу, або відразу кілька факторів - вибирається метод управління проектом, орієнтований на досягнення цього показника.

Найпоширенішим представником моделі системи розподілу завдань є таск-менеджер. Таск-менеджер – це спеціальне програмне забезпечення, яке допомагає формулювати та фіксувати завдання, контролювати хід роботи та закінчувати проекти вчасно.

Веб-застосунок – це комп’ютерна програма, яка використовує веб-браузер для виконання певної функції, інтерфейсом якої є веб-сайт. Веб-браузер в цьому випадку використовується, як клієнт, а веб-сервер в якості сервера. У порівнянні з локальним програмним забезпеченням, доступ до веб-застосунків можна отримати з будь-якого комп’ютера через Інтернет. Відповідно до цього даний таск-менеджер було розроблено у вигляді веб-застосунку з використанням технології, які використовуються в веб-розробці – JavaScript, HTML, CSS, VueJS, MongoDB, Node.js.

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		33

РОЗДІЛ 3

РЕАЛІЗАЦІЯ МОДЕЛІ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОПТИМІЗАЦІЇ ПРОЄКТІВ

3.1. Загальна характеристика програмного забезпечення

Програма Task Manager розроблена у вигляді веб-застосунку, для її використання необхідна лише наявність веб-браузера. Дана розробка пропонує приємний та інтуїтивно зрозумілий інтерфейс, для роботи з яким не потрібно проходити додаткове навчання, як для більшості аналогічних програм.

Програма включає в себе всі основні функції, які необхідні для організації роботи над проєктами: постановка завдань та керування ними, можливість спілкуватися з виконавцем та постановником безпосередньо в самому завданні, контроль виконання, додавання та видалення користувачів, доступ до списку користувачів з основною інформацією про них, перегляд стрічки новин компанії та додавання власних новин.

Task Manager містить додаткову функцію - «аналіз завантаженості працівника», яка аналізує навантаження на працівника і в разі перевищення допустимої норми, при постановці завдання, повідомляє про це постановника, а також може рекомендувати альтернативного виконавця для завдання.

Програма доступна на будь-якій платформі, за наявності встановленого сучасного веб-браузера та доступу до мережі Інтернет.

3.2. Модуль роботи з завданнями

На початку роботи з системою користувач потрапляє на сторінку авторизації (Рис. 3.1). На ній розміщена форма з полями для вводу логіну та паролю.

Зм.	Арк.	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02м- 20.МР</i>			
Зав. каф.		Криворучко О.В.		21.06.21	Модель інтелектуальної системи оптимізації проєктів ІТ компанії	Стадія	Арк.	Аркушів
Керівник		Савченко Т.В.		21.06.21		РЗ	34	50
Гарант		Токар В.В.		21.06.21		Факультет інформаційних технологій 2м курс, 2м група		
Розробив		Сологуб Л.В.		21.06.21				
					Реалізація моделі інтелектуальної системи оптимізації проєктів			

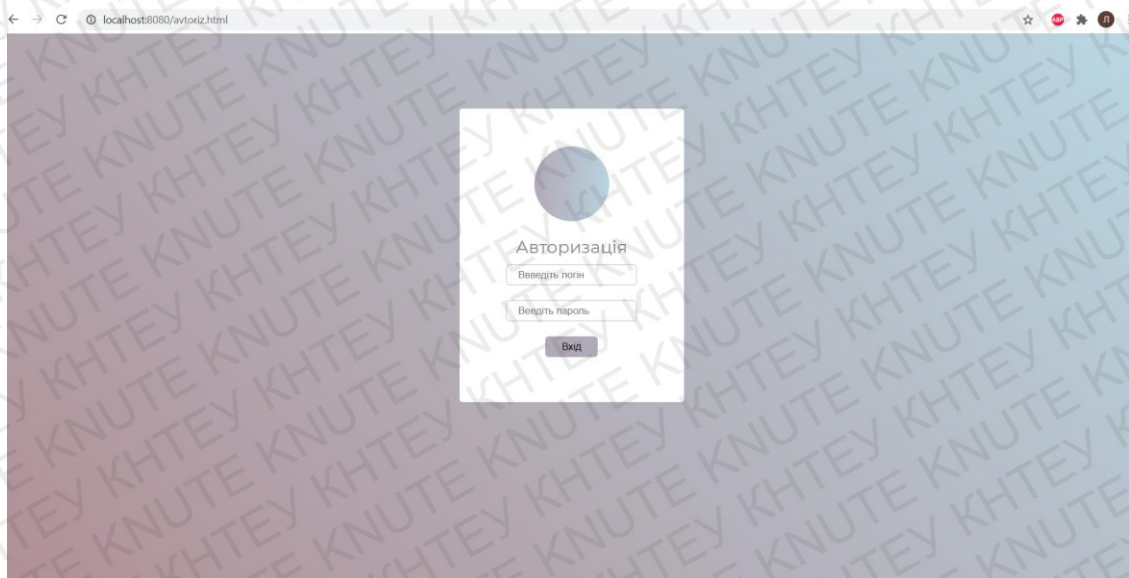


Рис. 3.1. Сторінка авторизації

Користувач має ввести логін та пароль для входу в систему. Якщо данні введено невірно, то буде виведено сповіщення про помилку (Рис. 3.2).

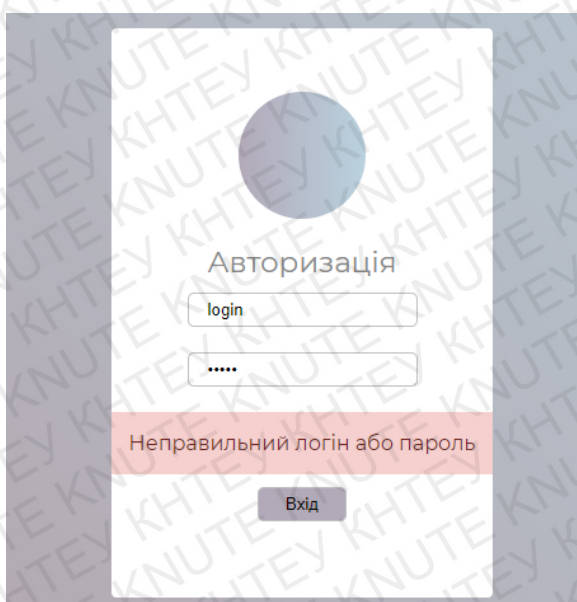


Рис. 3.2. Сповіщення про помилку

В разі успішної авторизації користувач отримує доступ до системи. В верхній частині сторінки відображається посада авторизованого користувача, поточний час та дата, фото та логін даного користувача, кнопка «Вийти», для

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		35

виходу з системи та перенаправлення на сторінку авторизації (Рис. 3.3)



Рис. 3.3 Верхня частина сторінки

З лівого боку розміщене скорочене бокове меню зі значками, які відповідають наступним пунктам: «Призначити завдання», «Мої завдання», «Працівники» та «Новини». Щоб розгорнути меню потрібно натиснути на кнопку зі стрілкою (Рис. 3.4).

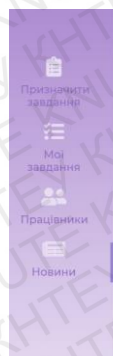


Рис. 3.4. Бокове меню

За замовчуванням, після авторизації, відкривається сторінка «Мої завдання» (Рис. 3.5). В даному розділі відображаються всі задачі, які потрібно виконати даному користувачу.

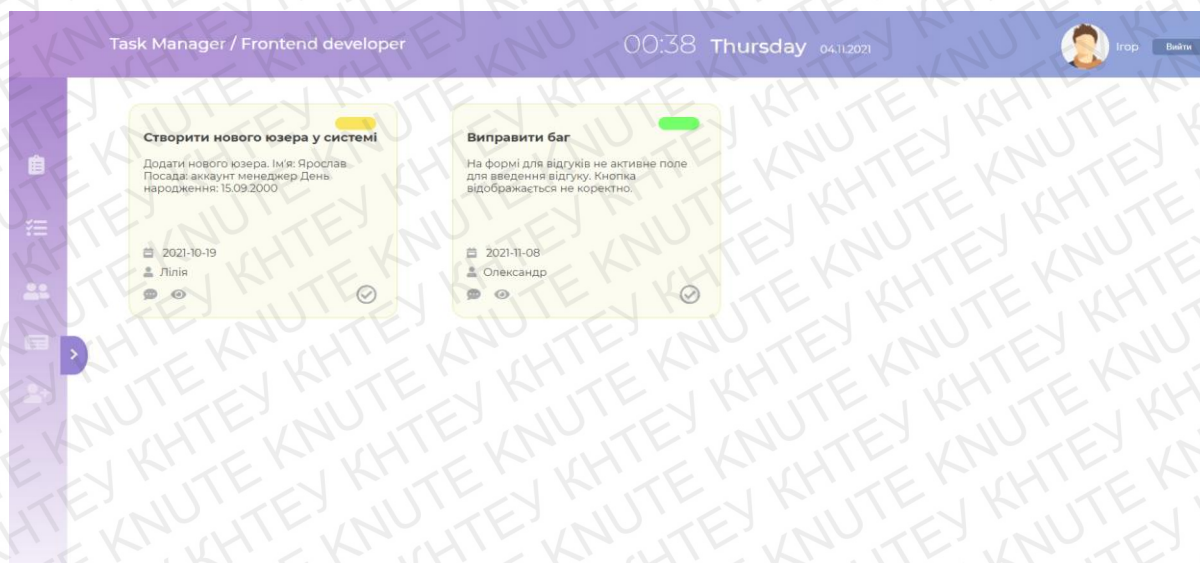


Рис. 3.5. Сторінка «Мої завдання»

					КНТЕУ 121 02м - 20.МР	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Дані про кожне завдання зберігаються в колекції *tasks* бази даних MongoDB (Рис. 3.6), в полях:

- *id* – унікальний ідентифікатор завдання;
- *creator* – ідентифікатор постановника завдання;
- *title* – назва завдання;
- *deadline* – дедлайн завдання;
- *complication* – складність завдання;
- *taskDescription* – опис завдання;
- *performer* – ідентифікатор виконавця;
- *status* – статус завдання.

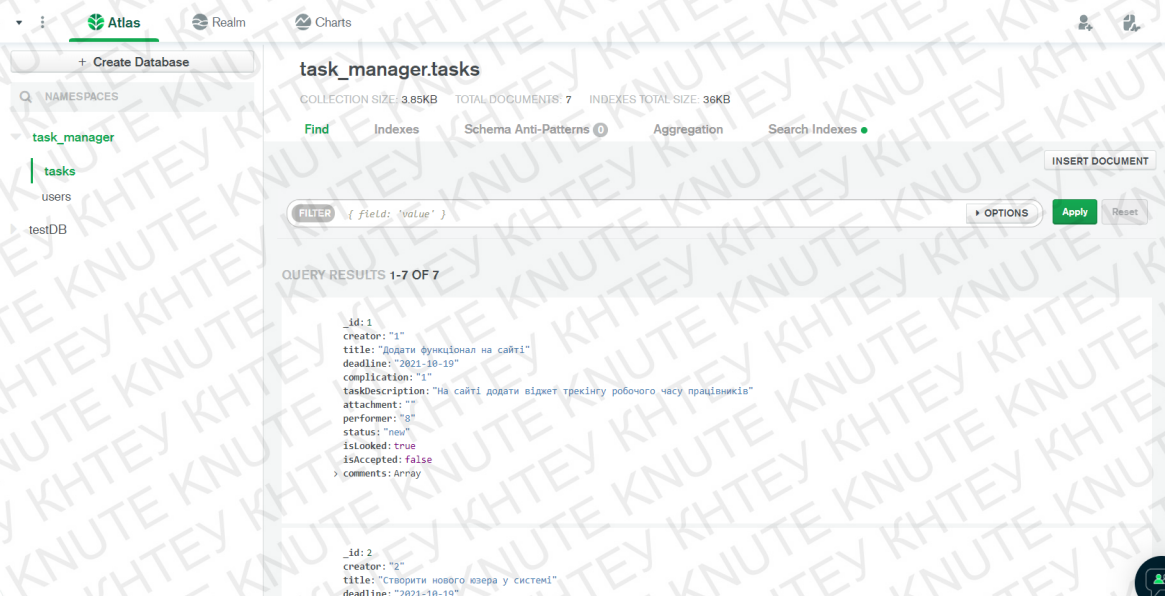


Рис. 3.6. Колекція *tasks*

На сторінці «Мої завдання» відображаються ті завдання, в яких ідентифікатор поточного користувача відповідає значенню поля *creator* бази даних.

Кожна картка завдання поділена на частини: маркер складності, який позначається відповідним кольором, що відповідає певному рівню складності (від 1 до 6), назва завдання, короткий опис, дедлайн, ім'я постановника, кнопки «Переглянути детальніше», «Коментувати» та «Завершити завдання» (Рис. 3.7).

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		37

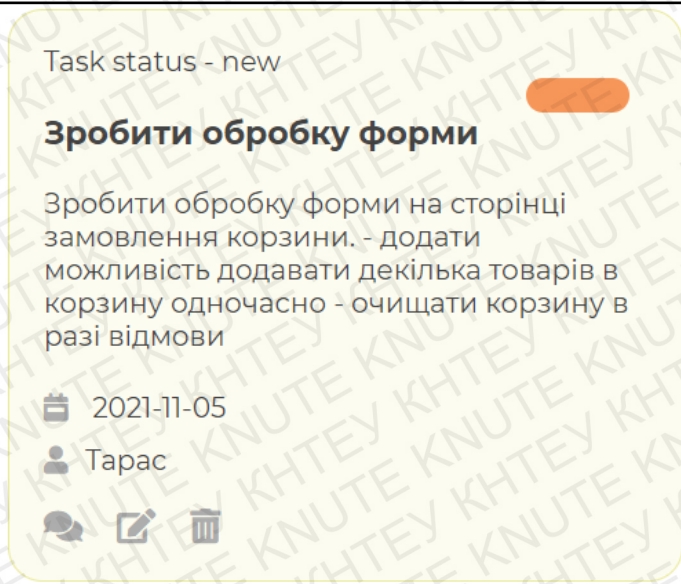


Рис. 3.7. Інтерфейс картки завдання

Після натиску на кнопку «Переглянути детальніше» відкривається форма з повною інформацією про відповідне завдання та кнопкою «Виконано», для завершення завдання.

Кнопка «Коментувати» відкриває форму коментарів. На ній відображаються написані раніше коментарі та є поле з кнопкою для введення нового (Рис. 3.8). Після натиснення кнопки «Додати», до даного завдання буде додано новий коментар та показано в формі.

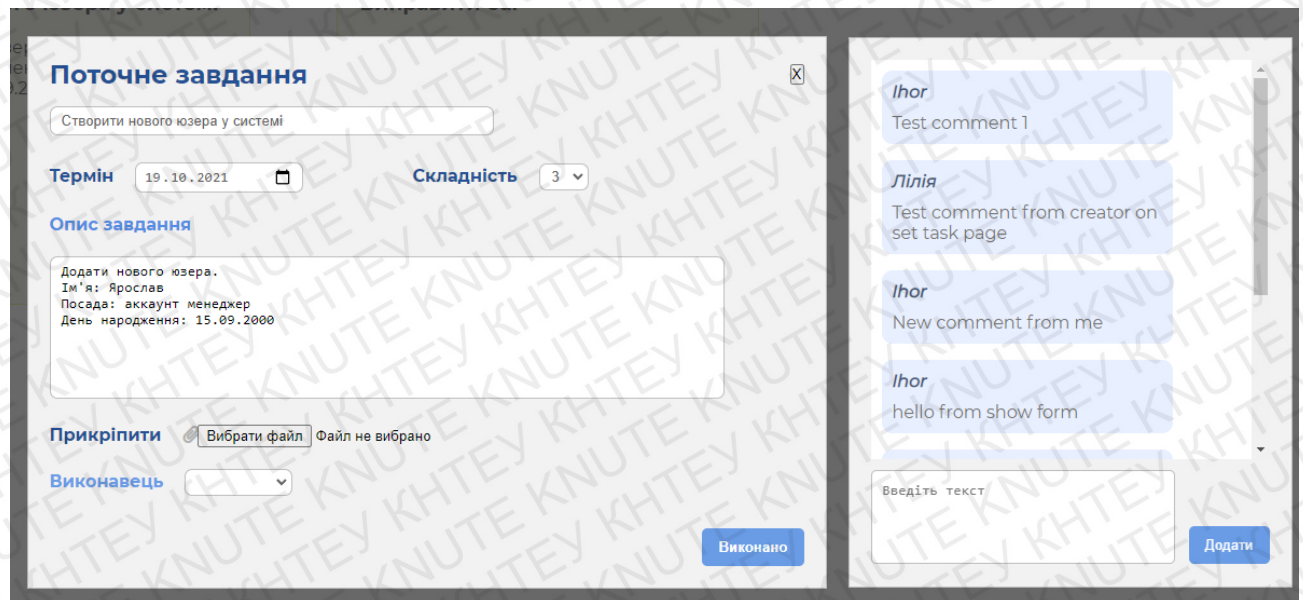


Рис. 3.8. Форма коментарів

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		38

Доступ до коментування завдань мають постановник та виконавець, які фігурують в них. Коментарі зберігаються в колекції *tasks*, в полі *comments*.

Кнопка «Завершити завдання», змінює статус завдання на виконане. Після цього воно більше не відображається в полі «Мої завдання».

Після переходу на сторінку «Призначити завдання», відображується кнопка «Призначити завдання», та список усіх активних задач, які були створені поточним користувачем (Рис. 3.9). Дані про завдання зберігаються в колекції *tasks* бази даних, і на сторінці відображаються ті завдання, де ідентифікатор працівника буде дорівнювати значенню поля *creator*.

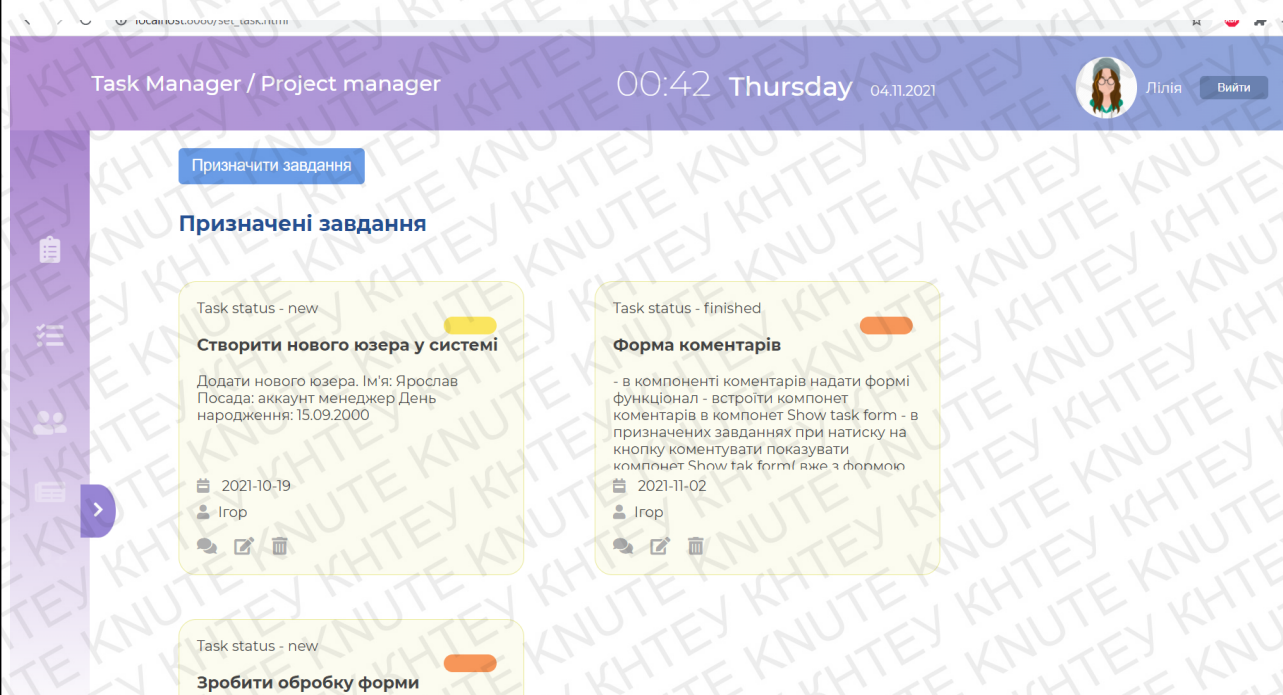


Рис. 3.9. Сторінка «Призначити завдання»

Інтерфейс картки завдання майже ідентичний до аналогічної картки зі сторінки «Мої завдання», відрізняється лише деякі кнопки: «Редагувати» та «Видалити завдання».

Кнопка «Призначити завдання» відкриває форму для створення нового завдання (Рис. 3.10). Дана форма має поля для введення назви задачі, терміну виконання, складності, детального опису, поле зі списком усіх працівників та кнопку «Призначити».

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		39

Після заповнення всіх полів форми, та натиснення кнопки, завдання з'являється в списку «Призначені завдання» і відображається зі статусом «нове» на сторінці «Мої завдання» у працівника, який був вказаний як виконавець (Рис. 3.12).

Призначені завдання

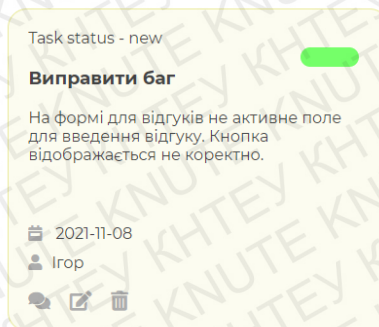


Рис. 3.12. Завдання відображається в призначених завданнях

Кнопка «Редагувати» на картці завдання, відкриває форму для редагування поточного завдання (Рис. 3.13). Після внесених змін і натиснення кнопки «Редагувати», дані обраного завдання оновлюються.

Рис. 3.13. Форма редагування завдання

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		41

Кнопка «Видалити завдання» видаляє обране завдання. Інформація про нього зникає з бази даних та відповідно не відображається на сторінці постановника та на сторінці «Мої завдання» у виконавця.

3.3. Модулі роботи з користувачами

При переході на сторінку «Працівники», відображається список зареєстрованих користувачів. Дані про всіх працівників зберігаються в колекції *users* бази даних (Рис. 3.14) в полях: *id* (унікальний ідентифікатор користувача), *surname* (прізвище користувача), *name* (ім'я користувача), *login* (логін користувача), *password* (пароль користувача), *email* (електронна адреса користувача), *phone* (номер телефону користувача), *date_of_birth* (дата народження користувача), *position* (посада користувача), *isAdmin* (права адміністратора), *image* (фото користувача), *employee_workload* (поточне навантаження на користувача).

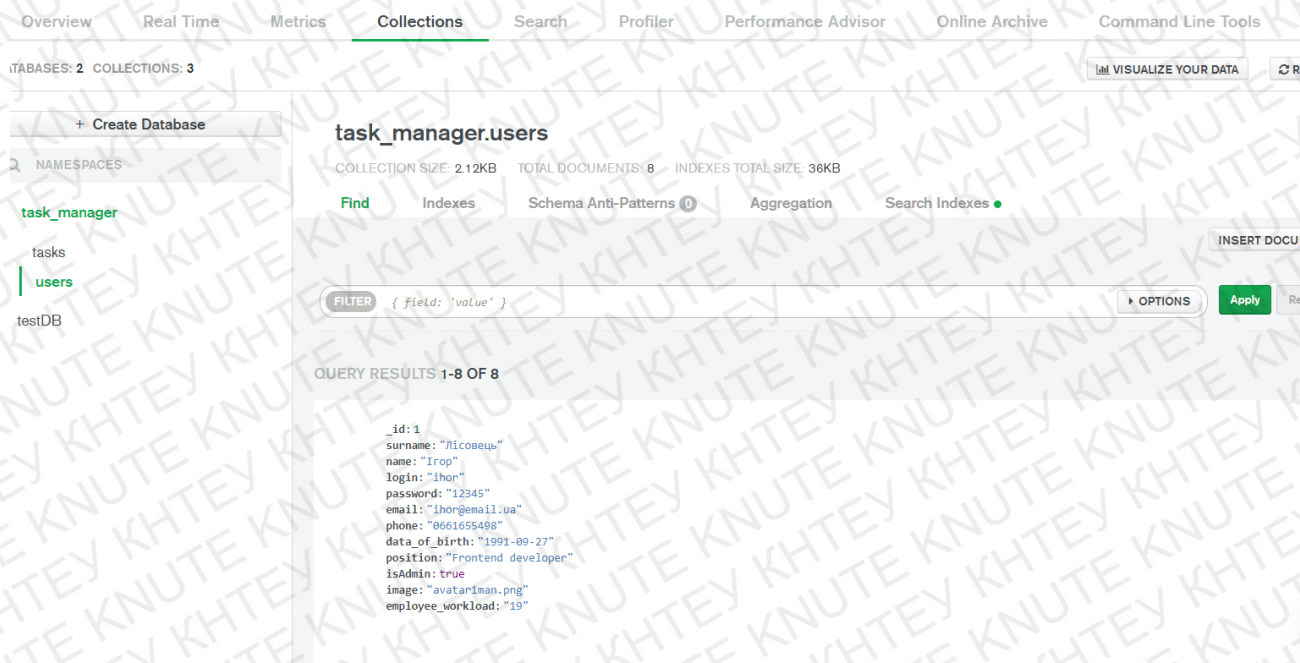


Рис. 3.14. Колекція *users*

На картці кожного працівника розміщене фото та основна інформація про

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		42

нього: ім'я та прізвище, посада, електронна пошта, номер телефону та дата народження (Рис. 3.15).

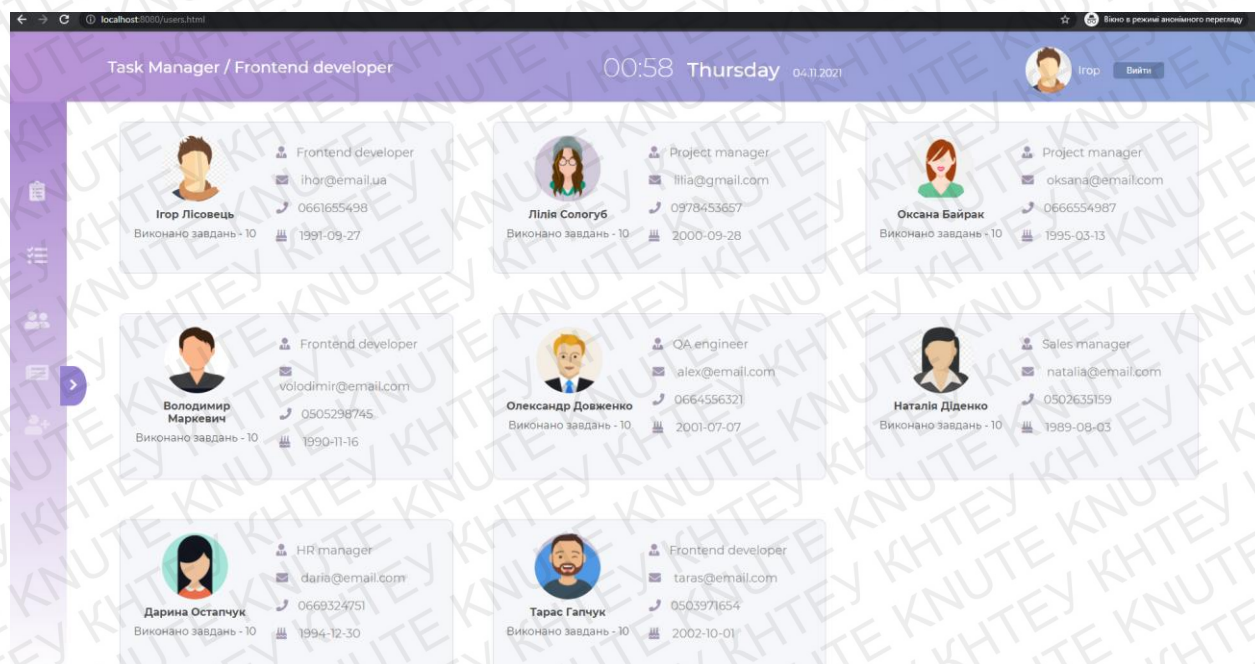


Рис. 3.15. Сторінка «Працівники»

Якщо користувач має права адміністратора, то на боковому меню з'являється кнопка для переходу в адміністративний розділ – «Додати користувача», яка для звичайного користувача є недоступною (Рис. 3.16).

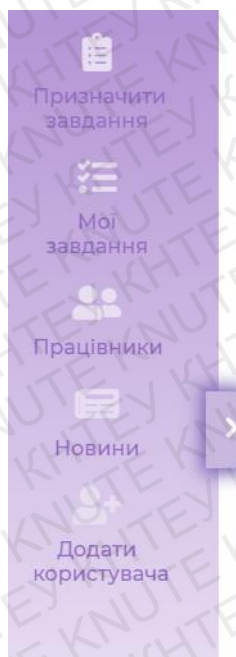


Рис. 3.16. Кнопка «Додати користувача»

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		43

При переході на цю сторінку відображається форма для додавання нового користувача та список користувачів (Рис. 3.17). Форма містить поля для введення прізвища, імені, логіну, паролю, електронної пошти, телефону, дати народження та посади. Також на ній є кнопка для завантаження фото та додавання користувача.

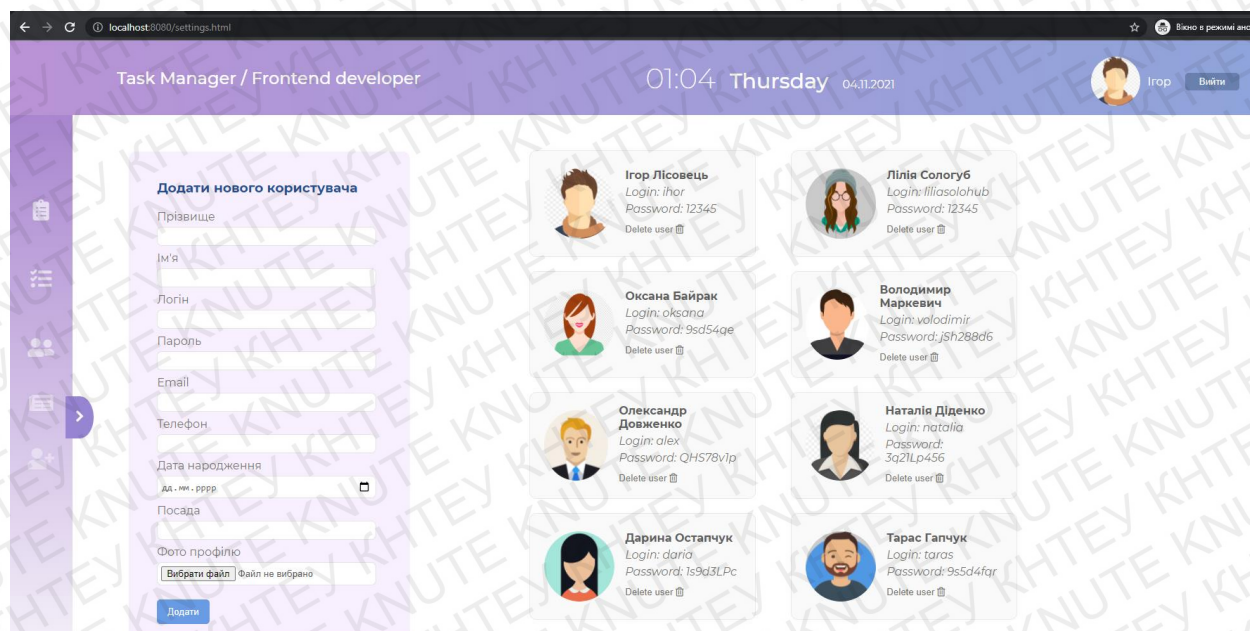


Рис. 3.17. Адміністративний розділ

Після того, як адміністратор заповнить усі поля і натисне кнопку «Додати», нового користувача буде додано в базу даних і він з'явиться в списку усіх користувачів.

Даний список складається з карток користувачів, дані отримуються з колекції *users* (Рис. 3.18).

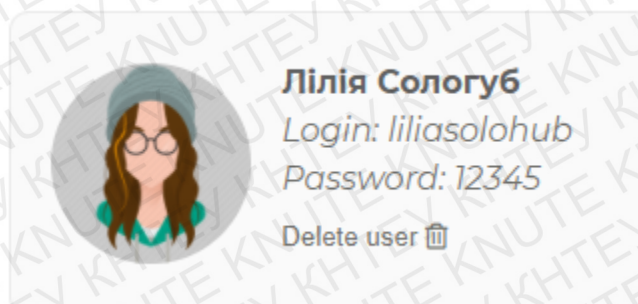


Рис. 3.18. Картка користувача

									Арк.
									44
Зм.	Арк.	№ докум.	Підпис	Дата					

На кожній з них розміщено фото, ім'я та прізвище, логін, пароль та кнопка «Видалити користувача», після натиску якої, користувач видалається з бази даних і відповідно більше не відображається в системі.

3.4. Додаткові функції

Додатковою функцією програми є сторінка «Новини». При переході на неї, відображаються стрічка новин компанії, форма для додавання власної новини, та віджет погоди (Рис. 3.19).

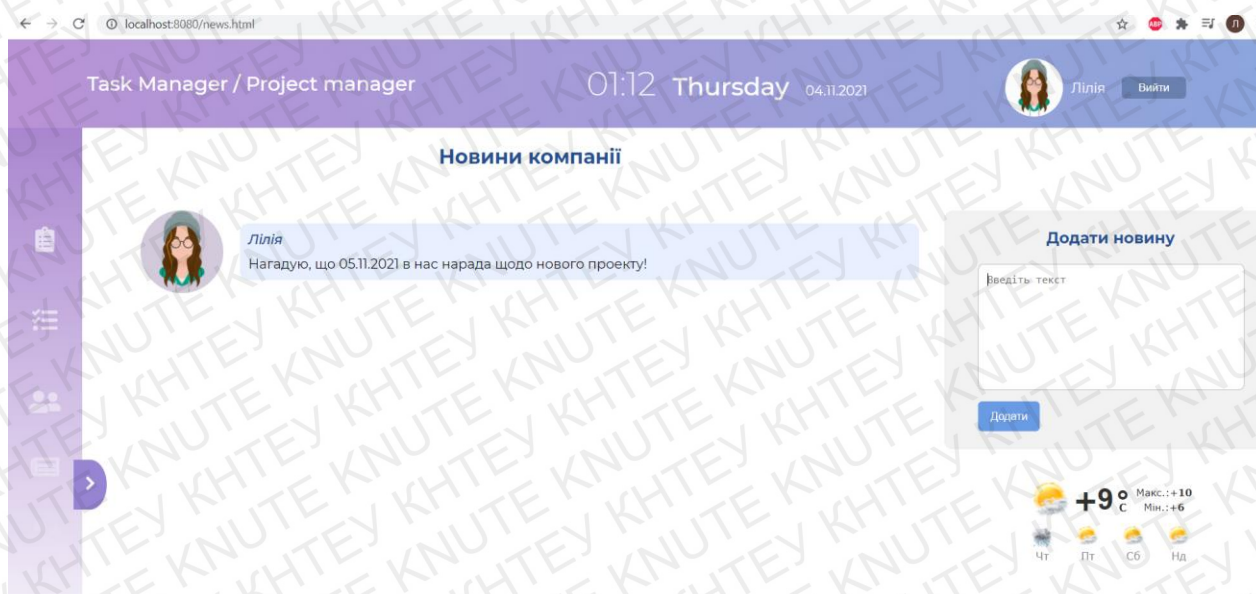


Рис. 3.19. Сторінка «Новини»

На стрічці новин відображається текст новин, дата публікації, імена та фото користувачів, що їх додали. Форма для створення власної новини містить поле для введення тексту та кнопку «Додати» (Рис. 3.20).

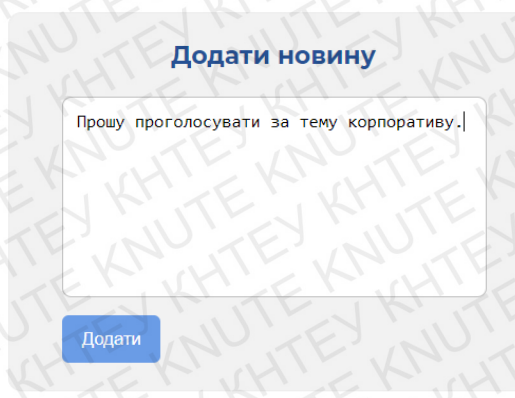


Рис. 3.20. Форма для додавання новини

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		45

Після заповнення форми та натиснення кнопки, новина відображається в стрічці (Рис.3.21). Кожне нове повідомлення відображається над попередніми, таким чином користувачі відразу можуть побачити актуальні новини.

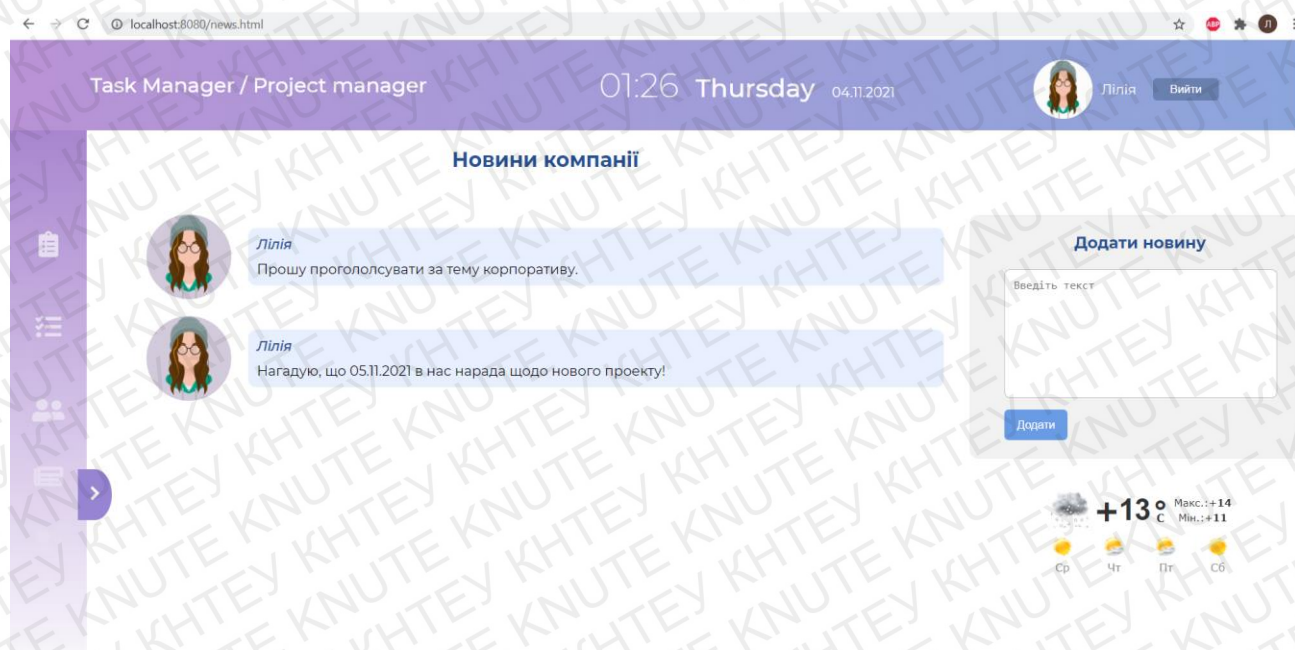


Рис. 3.21. Додана новина відображається в стрічці

3.5. Висновки до розділу 3

Отже, розроблений веб-застосунок є кросплатформним, має приємний та інтуїтивно зрозумілий інтерфейс, є зручним у використанні, не потребує спеціальної підготовки користувачів, містить в собі основні функції для оптимізації організації роботи над проектами та автоматичний аналіз завантаженості працівників.

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		46

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання дипломного проекту було проведено теоретичні дослідження, розроблено модель інтелектуальної системи оптимізації проектів ІТ компанії та отримано такі висновки:

1. На сьогоднішній день багато компаній працюють над складними, дороговартісними проектами, робота над якими вимагає великих затрат часу та уваги з боку працівників. Робота над проектами - дуже кропіткий процес, особливо над великими і складними. Ключем до успішного і своєчасного виконання будь-якого проекту є грамотно організована робота над ним. Тому все більш актуальним стає використання засобів, які мають за мету допомогти в плануванні та контролюванні процесів, які виникають під час роботи над проектами.
2. Програмне забезпечення для управління завданнями допомагає компаніям оптимізувати життєвий цикл проекту, мінімізувати можливості раптових перешкод та сприяти комунікації та співпраці між командами, що заощаджує час та гроші.
3. В процесі виконання проекту розроблено та реалізовано модель інтелектуальної системи оптимізації проектів в вигляді веб-застосунку.
4. Розроблений веб-застосунок є кросплатформним, має приємний та інтуїтивно зрозумілий інтерфейс, є зручним у використанні, не потребує спеціальної підготовки користувачів, містить в собі основні функції для оптимізації організації роботи над проектами та автоматичний аналіз завантаженості працівників.

Зм.	Арк.	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02м- 20.МР</i>			
Зав. каф.		Криворучко О.В.		01.11.21	Модель інтелектуальної системи оптимізації проектів ІТ компанії	Стадія	Арк.	Аркушів
Керівник		Савченко Т.В.		01.11.21		ВП	47	50
Гарант		Токар В.В.		01.11.21		Факультет інформаційних технологій		
Розробив		Сологуб Л.В.		01.11.21		2м курс, 2м група		
					<i>Висновки та пропозиції</i>			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ременяк Л.В. Управління IT-проектами: конспект лекцій. Одеса, Одеський державний екологічний університет, 2015. 168 с.
2. Офіційний сайт Capterra [Електронний ресурс]. – Режим доступу: <https://www.capterra.com>
3. Катренко А.В. Управління IT-проектами. [Книга 1. Стандарти, моделі та методи управління проектами] : [підручник]. – Львів: «Новий Світ - 200».
4. Гринчак М.В., Шаповалов А.Л., Кузьмичова К.В., Волков Д.О. Інформаційні системи й технології на підприємстві: Конспект лекцій (для студентів 4 курсу заочної форми навчання за напрямом підготовки 0501 “Економіка і підприємництво” спеціальності “Економіка підприємства”) / Харків: ХНАМГ, – 2009. 84с.
5. IEEE Std 1074-1995, IEEE Standard for Developing Software Life Cycle Processes
6. The Differences Between IT Projects And Other Types of Projects [Електронний ресурс]. – Режим доступу: <https://scrumteam.pro/the-differences-between-it-projects-and-other-types-of-projects-and-how-project-managers-should-deal-with-them/>
7. «Комп’ютерні науки. Управління проектами» / І.А. Чуб, М.В. Новожилова, М.П. Пан; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків: ХНУМГ ім. О.М. Бекетова, 2018. – 119 с.
8. Коцовський В. М. Інтелектуальні інформаційні системи: конспект лекцій. Ужгород, Ужгородський національний університет, 2019. 73 с.
9. Key Benefits of Using a Task Management Software [Електронний ресурс]. – Режим доступу: <https://weekplan.net/benefits-of-task-management-software>
10. Таск-менеджери: що до чого і як обрати саме свій [Електронний ресурс]. –

					<i>КНТЕУ 121 02м- 20.МР</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		27.02.21	Модель інтелектуальної системи оптимізації проектів ІТ компанії	<i>Стадія</i>	<i>Арк.</i>	<i>Аркушів</i>
Керівник		Савченко Т.В.		27.02.21		<i>СВД</i>	<i>48</i>	<i>50</i>
Гарант		Токар В.В.		27.02.21		<i>Факультет інформаційних технологій</i> <i>2м курс, 2м група</i>		
Розробив		Сологуб Л.В.		27.02.21				
					<i>Список використаних джерел</i>			

Режим доступу: <https://studway.com.ua/task-menedzheri/>

11. Офіційний сайт Jira [Електронний ресурс]. – Режим доступу: <https://www.atlassian.com/software/jira>

12. Офіційний сайт Basecamp [Електронний ресурс]. – Режим доступу: <https://basecamp.com/>

13. Старченко Г. В. Управління проектами: теорія та практика : навч. посіб. / Г. В. Старченко. – Чернігів : видавець Брагинець О. В., 2018. – 306 с.

14. Методології управління проектом [Електронний ресурс]. – Режим доступу: <https://www.staff.ua/uk/bloh/metodologiji-upravlinnya-proektom>

15. The benefits of web applications in todays technological era [Електронний ресурс]. – Режим доступу: <https://www.kcsitglobal.com/blogs/detail-blog/the-benefits-of-web-applications-in-todays-technological-era>

16. The benefits of using web-based applications [Електронний ресурс]. – Режим доступу: <https://www.geeks.ltd.uk/insights/the-benefits-of-using-web-based-applications>

17. What is JavaScript used for? [Електронний ресурс]. – Режим доступу: <https://www.hackreactor.com/blog/what-is-javascript-used-for>

18. What Is HTML? Hypertext Markup Language Basics Explained [Електронний ресурс]. – Режим доступу: <https://www.hostinger.com/tutorials/what-is-html>

19. Flanagan, David. JavaScript – The definitive guide (6 ed.). p. 1. JavaScript is part of the triad of technologies that all Web developers must learn: HTML to specify the content of web pages, CSS to specify the presentation of web pages, and JavaScript to specify the behavior of web pages.

20. "What is CSS?". World Wide Web Consortium. Archived from the original on 2010-11-29. Retrieved 2010-12-01. [Електронний ресурс]. – Режим доступу: <https://www.w3.org/standards/webdesign/htmlcss#whatcss>

21. "Web-based Mobile Apps of the Future Using HTML 5, CSS and JavaScript". HTMLGoodies. 23 July 2010. Archived from the original on 2014-10-20.

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		49

22. "Reactivity in Depth — Vue.js" [Електронний ресурс]. – Режим доступу: <https://vuejs.org/v2/guide/reactivity.html>
23. Офіційний сайт MongoDB [Електронний ресурс]. – Режим доступу: <https://searchdatamanagement.techtarget.com/definition/MongoDB>
24. About Node.js, and why you should add Node.js to your skill set? [Електронний ресурс]. – Режим доступу: <http://blog.training.com/2016/09/about-nodejs-and-why-you-should-add.html>
25. Інформаційні технології та моделювання бізнес-процесів : навч. посіб. / О. М. Томашевський, Г. Г. Цегелик, М. Б. Вітер, В. І. Дубук. - К. : ЦУЛ, 2012. - 296 с.
26. Нестеренко О.В., Ковтунець О.В., Фаловський О.О. Інтелектуальні системи і технології. Ввідний курс: Навч. Посібник. – К.: Національна академія управління, 2017. – 90 с.

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>50</i>

ТЕХНІЧНЕ ЗАВДАННЯ

1. Загальні відомості

1.1. Найменування системи

Task Manager

1.2. Планові терміни початку та закінчення робіт

Початок: 21.09.2020

Закінчення: 25.11.2021

1.3. Порядок оформлення і пред'явлення результатів робіт

1.3.1. Структура робіт проекту

1.3.2. Детальний план робіт

1.3.3. Матриця розподілу відповідальності в проекті

1.3.4. Реєстр ризиків проекту

1.3.5. Проектна документація

1.3.6. Виконання робіт по етапам проекту та проекту в цілому

1.3.1. Виконавець формує докладний перелік робіт щодо впровадження системи за основними напрямками для кожної функціональної області згідно даного переліку вимог:

- Управління проектом
- Документація
- Користувачі
- Процеси
- Тестування

1.3.2. Виконавець формує та узгоджує із Замовником детальний план робіт по проекту. Планування робіт виконується за методологією, що передбачена виробником програмної платформи та Виконавцем.

1.3.3. Виконавець формує та узгоджує матрицю відповідальності (RASCI),

Зм.	Арк.	№ докум.	Підпис	Дата	КНТЕУ 121 02м- 20.МР			
Зав. каф.		Криворучко О.В.		20.03.21	Модель інтелектуальної системи оптимізації проєктів ІТ компанії	Стадія	Арк.	Аркушів
Керівник		Савченко Т.В.		20.03.21		ТЗ	51	50
Гарант		Токар В.В.		20.03.21		Факультет інформаційних технологій 2м курс, 2м група		
Розробив		Сологуб Л.В.		20.03.21				
					Технічне завдання			

що описує розподіл ключових робіт проекту між ролями/учасниками проектної команди.

1.3.4. Виконавець готує та узгоджує реєстр ризиків проекту, у якому надається:

- опис можливих ризиків
- вірогідність виникнення ризиків
- стратегія запобігання, мінімізації та нівелювання ризиків.

1.3.5. Виконавець готує перелік та шаблони проектної документації, що містять описи всіх необхідних налаштувань та інформацію, необхідну для впровадження системи, а саме:

- Детальний опис бізнес-вимог
- Опис реалізації бізнес-вимог
- Функціональну та технічну архітектуру
- Опис налаштувань
- Опис доопрацювань системи (функціональний дизайн)
- Опис доопрацювань системи (технічний дизайн)
- Опис конвертації даних
- Сценарії тестування функціональності
- Сценарії тестування потужності
- Документацію для навчання адміністраторів та користувачів
- Опис структури колекцій БД
- Інструкції для користувачів

1.3.6. Виконання робіт по етапах проекту затверджується актами виконаних робіт.

Прийманню Замовником підлягають результати робіт виконання етапів. Про свою готовність до здачі робіт Виконавець письмово повідомляє Замовника.

1.4. Головним бенефіціаром виступає КНТЕУ.

					<i>КНТЕУ 121 02м - 20.МР</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		52

2. Мета та призначення створення системи

2.1. Система призначена для оптимізації роботи над проектами ІТ компанії.

2.2. Метою створення системи є можливість ефективного розподілу задач між працівниками компанії спираючись на автоматичний аналіз їх завантаженості

3. Вимоги до системи

3.1. Вимоги до системи в цілому

- Повинна розроблятися на базі сучасних інформаційних систем різних класів як блоків функціональності, кожен з яких відповідає затвердженим функціональним вимогам та успішно інтегрується з іншими блоками функціональності для ефективного використання із застосуванням найкращих світових практик.

- Повинна бути побудована за клієнт-серверною архітектурою.

- Повинна надавати доступ користувачам до інтерфейсу системи за допомогою браузера через існуючу internet-мережу.

3.1.1. Вимоги до структури та функціонування системи, перелік підсистем

3.1.1.1. Вимоги до способів і засобів інформаційного обміну між компонентами системи

Система повинна отримувати та опрацьовувати дані засобами REST API отримані з інтерфейсу користувача та сервера і зберігати їх локально в базі даних для подальшого використання цих даних безпосередньо системою.

3.1.1.2. Вимоги до режимів функціонування системи

Система повинна функціонувати в режимі онлайн.

3.1.2. Вимоги до пристосованості системи до змін

Забезпечення пристосованості системи повинно виконуватися за рахунок:

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		53

- своєчасності адміністрування;
- модернізації процесів збору, обробки і збереження даних відповідно до нових вимог;
- модифікації процедур доступу і представлення даних кінцевим користувачам;

3.1.3. Вимоги до надійності

Вимоги до надійності технічних засобів і програмного забезпечення визначаються їх розробниками.

Оцінка надійності та стабільності роботи буде оцінюватися за наступними параметрами: Uptime/Availability – безперервний час роботи або доступності системи (визначається у відсотках, наприклад 99.99%).

Основні показники надійності роботи системи будуть вимірюватись на етапі тестування та дослідної експлуатації системи. Метод дослідження та оцінки – експериментальний, тобто в тестовому режимі мають бути відтворені основні аварійні ситуації, за яких Опис вимог мають бути підтверджені показники надійності та відновлення працездатності системи.

3.1.4. Вимоги до ергономіки та технічної естетики

Комунікаційне, серверне та інше обладнання в ході виконання робіт не виготовляється і вимоги щодо ергономіки та технічної естетики забезпечуються відповідними Виробниками обладнання, що використовується.

3.1.5. Вимоги до інформаційної безпеки

1. Можливість встановлення захищеного з'єднання між клієнтом та сервером.
2. Наявність механізмів контролю за остаточним видаленням даних.
3. Рольова модель доступу користувачів до ресурсів системи.
4. Запобігання несанкціонованого внесення змін або знищення баз даних.

3.1.6. Вимоги до захисту від впливу зовнішніх факторів

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		54

Для роботи системи має бути забезпечено такі режими резервування даних, система повинна їх підтримувати:

1. Живе/постійне резервування даних на рівні системи зберігання даних.
2. Регулярне резервування (резервування даних програмними засобами у визначені інтервали).

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>55</i>

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

Необхідно виконати тестування веб-застосунку додатку. Для успішного проведення тестування, слід виконати наступні тест-кейси:

1. Робота зі сторінкою авторизації
2. Робота зі сторінкою поставлених задач
3. Робота з адміністративною сторінкою
4. Робота зі сторінкою новин

Кроки для виконання тест-кейсів та очікуваний результат описані в

Таблиця 6.1.

ПОРЯДОК ТЕСТУВАННЯ

Тест-кейс	Кроки для виконання	Очікуваний результат
Робота зі сторінкою авторизації	Вхід до системи з використанням правильних даних	Буде відкрита домашня сторінка у веб-застосунку
	Вхід до системи з використанням неправильних даних	На сторінці авторизації буде відображена помилка
Робота зі сторінкою поставлених задач	Створення нової задачі	Задача з'явиться в базі даних та відобразиться на сторінці задач постановника та виконавця
	Видалення задачі	Вибрана задача буде видалена з бази даних та зі сторінок поставлених та отриманих задач

					<i>КНТЕУ 121 02м- 20.МР</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Модель інтелектуальної системи оптимізації проєктів ІТ компанії <i>Програма та методика тестування</i>	<i>Стадія</i>	<i>Арк.</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		18.10.21		ПМТ	56	50
Керівник		Савченко Т.В.		18.10.21				
Гарант		Токар В.В.		18.10.21				
Розробив		Сологуб Л.В.		18.10.21				
					Факультет інформаційних технологій 2м курс, 2м група			

Продовження таблиці 6.1.

	Редагування задачі	Поля вибраної задачі будуть змінені в базі даних та на сторінці задач
Тест-кейс	Кроки для виконання	Очікуваний результат
Робота з адміністративною сторінкою	Додавання нового користувача	Новий користувач буде доданий в базу даних та відобразиться у спискові усіх користувачів
	Видалення користувача	Вибраний користувач буде видалений з бази даних та зникне зі списку усіх користувачів
Робота зі сторінкою новин	Додавання новини	Нова новина буде додана в базу даних та з'явиться на сторінці новин компанії

Якщо результати виконання тест-кейсів будуть аналогічними очікуваному результату, тестування можна вважати успішним.

					КНТЕУ 121 02м - 20.МР	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

КЕРІВНИЦТВО КОРИСТУВАЧА

В веб-застосунку для користувача доступні для використання наступні можливості:

1. постановка задачі з детальним її описанням(назва, детальний опис, дедлайн, складність, виконавець, завантаження файлів).
2. керування поставленими та отриманими задачами: перегляд, коментування, редагування, підтвердження виконання та видалення задачі.
3. перегляд усіх зареєстрованих користувачів та основної інформації про них.
4. перегляд та створення новин.
5. для користувачів зі статусом «адмін» можливість додавати та видаляти користувачів.

Для того, щоб почати користуватися можливостями потрібно пройти етап авторизації. Для цього на екранові авторизації потрібно ввести логін та пароль користувача.

Після успішної авторизації доступ до системи буде наданий. Відкривається сторінка з отриманими задачами. Також відображається бокова панель навігації та верхня панель. На боковій панелі наявні кнопки для переходу на сторінку створення задач та перегляду створених користувачем задач – «**Призначити завдання**», отриманих задач користувачем – «**Мої завдання**», сторінку зі списком усіх працівників – «**Працівники**» та на сторінку новин компанії – «**Новини**». Для адміністраторів також наявна додаткова можливість переходу на адміністративну сторінку – «**Додати користувача**».

Зм.	Арк.	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		21.10.21	Модель інтелектуальної системи оптимізації проектів ІТ компанії Керівництво користувача	Стадія	Арк.	Аркушів
Керівник		Савченко Т.В.		21.10.21		КК	58	50
Гарант		Токар В.В.		21.10.21		Факультет інформаційних технологій		
Розробив		Сологуб Л.В.		21.10.21		2м курс, 2м група		

КНТЕУ 121 02м- 20.МР

1. На сторінці «**Призначити завдання**» для створення нової задачі потрібно натиснути кнопку «**Призначити завдання**». Після цього необхідно заповнити наступні поля задачі: назва задачі, термін, складність, опис завдання та виконавець. Після цього натиснути кнопку «**Призначити**». Файл додається опціонально.

2. На сторінці «**Мої завдання**» відображається список задач створених поточним користувачем та власне кнопка створення нової задачі. Створені задачі користувач може коментувати, редагувати та видаляти при натисканні на відповідні кнопки. Коментарі до задачі вносяться в поле форми, яка розташована праворуч від картки задачі. Для відправки коментаря потрібно натиснути кнопку «**Додати**»

3. На сторінці «**Працівники**» відображається список усіх користувачів з короткою інформацією про них.

4. На сторінці «**Новини**» відображається список усіх новин компанії, а також вікно для публікації власної новини. Для публікації новини потрібно ввести текст новини в поле «**Додати новину**» та натиснути кнопку «**Додати**». Після цього новина відобразить на сторінці

5. На сторінці «**Додати користувача**» користувачі зі статусом «адмін» можуть додавати та видаляти користувачів. Для додавання користувача необхідно заповнити усі поля форми «**Додати нового користувача**». А саме: *Прізвище, ім'я, логін, пароль, Email, телефон, дата народження, посада* та додати фото профілю. Після цього натиснути кнопку «**Додати**». Після цього новий користувач з'явиться у загальному спискові.

Для видалення користувача потрібно натиснути кнопку «**Delete user**». Після натискання вибраний користувач буде видалений із системи.

					<i>КНТЕУ 121 02м - 20.МР</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		59

ДОДАТКИ

Додаток А

Лістинг програмного коду для створення завдань

```
<template>
  <div id="set_form" class="form_bg">
    <div class="set_task_form">
      <!-- <p>Set task form</p> -->
      <form id="resForm" action="api/createTask" method="post" class="form">
        <div class="set_task_form__header">
          <h2>Нове завдання</h2>
          <input name="creator" v-bind:value='creator_id' style="display:none">
          <button class="set_task_form__close_btn" @click="closeFormBtnClick">X</button>
        </div>
        <input type="text" class="set_task_form__name" placeholder="Назва задачі" name="title" v-
model='taskTitle'>
        <div class="set_task_form__details">
          <div class="set_task_form__deadline">
            <span>Термін</span>
            <input type="date" id="deadline" name="deadline" v-model='taskDeadline'>
          </div>
          <div class="set_task_form__complication">
            <span>Складність</span>
            <select name="complication" id="complication" v-model='taskComplexity'>
              <option value="1">1</option>
              <option value="2">2</option>
              <option value="3">3</option>
              <option value="4">4</option>
              <option value="5">5</option>
              <option value="6">6</option>
            </select>
          </div>
        </div>
        <div class="set_task_form__description">
          <p>Опис завдання</p>
          <textarea name="taskDescription" rows="8" cols="80" v-model='taskDescription'></textarea>
        </div>
        <div class="set_task_form__attach">
```

Продовження дод. А

```
<span>Вкладення</span>
<div class="set_task_form__attachent">
  <i class="fas fa-paperclip"></i>
  <input type="file" name="attachment">
</div>
</div>
<div class="set_task_form__performer">
  <span>Виконавець</span>
  <select name="performer" id="performer" v-model='taskPerformer'>
    <option v-for="user in allUsers" v-bind:key="user._id" v-
bind:value="user._id">{{ user.name }}</option>
  </select>
</div>
<div class="set_task_form__assign">
  <div v-if="checkEmployeeWorkload > 20" class="assign_notice">Завдання такої складності
перенавантажить працівника, і може бути виконане не вчасно. <p> Для виконання даного завдання Ви
можете обрати іншого. Наприклад - {{ getAltertnativeUser }}</p></div>
  <button id="set_task_form__assign_btn" v-if='!edit'>Призначити</button>
<button id="set_task_form__assign_btn" v-else type="button" @click='editTask'
onclick="event.preventDefault()">Редаргувати</button>
</div>
  <input name="new_employee_workload" :value="checkEmployeeWorkload" style="display: none">
</form>
</div>
</div>
</template>
<script>
import { busS } from '../entry/set_task.js';
import $ from 'jquery'
export default {
  name: 'SetTaskForm',
  data() {
    return {
      taskId: "",
      taskTitle: "",
      taskDeadline: "",
      taskComplexity: "",
```

Продовження дод. А

```
taskDescription: ",
taskPerformer: ",
creator_id: localStorage.getItem("id"),
edit: false,
readOnly: false
}
},
created() {
  bus.$on('editBtnClick', data => {
    this.taskId = data._id,
    this.taskTitle = data.title;
    this.taskDeadline = data.deadline;
    this.taskComplexity = data.complication;
    this.taskDescription = data.taskDescription;
    this.taskPerformer = data.performer;
    this.edit = data.edit;
  })
},
updated() {
  $("#set_task_form__edit_btn").on('click', function() {
    $(".form_bg").removeClass("flex");
  })
},
props: {
  allUsers: Array
},
methods: {
  editTask: async function(){
    this.edit = false;
    let taskToUpdate = { "_id": this.taskId, "creator": this.creator_id, "title": this.taskTitle, "deadline":
this.taskDeadline,
    "complication": this.taskComplexity, "taskDescription": this.taskDescription, "attachment": "",
    "performer": this.taskPerformer}
    console.log(taskToUpdate);
    // const response =
    await fetch(`/api/updateTask`, {
      method: 'POST',
```

Продовження дод. А

```
headers: {'Content-Type': 'application/json'},
body: JSON.stringify(taskToUpdate)
})
document.location.reload()
},
closeFormBtnClick(e) {
e.preventDefault();
this.taskId = "";
this.taskTitle = "";
this.taskDeadline = "";
this.taskComplexity = "";
this.taskDescription = "";
this.taskPerformer = "";
this.edit = false;
}
},
computed: {
checkEmployeeWorkload() {
if (this.taskPerformer === "") {
return 0;
}
let taskPerformer = this.allUsers[this.taskPerformer-1];

if (parseInt(this.taskComplexity) + parseInt(taskPerformer.employee_workload) > 20) {
return parseInt(this.taskComplexity) + parseInt(this.allUsers[this.taskPerformer-1].employee_workload);
} else {
return parseInt(this.taskComplexity) + parseInt(this.allUsers[this.taskPerformer-1].employee_workload);
}
},
getAlternativeUser() {
if (this.taskPerformer === "") {
return;
}
let taskPerformer = this.allUsers[this.taskPerformer-1];
let taskPerformerPosition = taskPerformer.position;
```

Продовження дод. А

```
if (parseInt(this.taskComplexity) + parseInt(taskPerformer.employee_workload) > 20) {  
  let alternativeUser = this.allUsers.filter( (item) => {  
    return (item.position === taskPerformerPosition && item._id !== this.taskPerformer &&  
(parseInt(item.employee_workload) <= 20 - parseInt(this.taskComplexity)));  
  })  
  let namesOfAltUsers = [];  
  alternativeUser.forEach((item) => {  
    namesOfAltUsers.push(item.name)  
  });  
  return namesOfAltUsers.join(', ');  
  // return alternativeUser;  
} else {  
  return "";  
}} }  
</script>
```

Лістинг програмного коду для створення задачі в базі даних

```
app.post('/api/createTask', urlencodedParser, function(req, res) {
  if (!req.body) return res.sendStatus(400);
  console.log(req.body);
  let userObj = {'id': parseInt(req.body.performer), 'new_employee_workload':
req.body.new_employee_workload}
  delete req.body.new_employee_workload
  console.log(req.body);
  console.log(userObj);
  async function callAddTaskPost(req){
    await updateUser(userObj);
    await addTaskPost(req);
    res.redirect("/set_task.html")
  }
  callAddTaskPost(req)
});
async function addTaskPost(req) {
  const uri =
"mongodb+srv://admin:admin@cluster0.uvxe1.mongodb.net/task_manager?retryWrites=true&w=majority";
  const client = new MongoClient(uri);
  const taskToAdd = req.body;
  try {
    await client.connect();
    const collection = client.db("task_manager").collection("tasks");
    taskToAdd._id = await collection.countDocuments() + 1;
    taskToAdd.status = 'new'; // до задачі додається статус "new"
    taskToAdd.isLooked = false; // чи задача переглянута виконвцем
    taskToAdd.isAccepted = false; // чи задача прийнята постановником
    taskToAdd.comments = []; // масив для коментарів
    const taskArr = await collection.insertOne(taskToAdd);
    console.log(taskArr);
  } catch (e) {
    console.error(e);
  } finally {
    await client.close();
  }
}
```