

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ СПОВІЩАННЯ ДЛЯ
ПОКРАЩЕННЯ РОБОТИ ВІЙСЬККОМАТУ З ВИКОРИСТАННЯМ ОБ'ЄКТНО-
ОРІЄНТОВНИХ МОВ ПРОГРАМУВАННЯ»

Виконав: студент 4 курсу, групи ПД-42
спеціальності 121 Інженерія програмного
забезпечення

Бондар В.В.

Керівник Негоденко О.В.

Рецензент _____

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність підготовки 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерія програмного забезпечення

_____ Негоденко О.В.

« _____ » _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА
БОНДАРЮ ВАСИЛЮ ВІКТОРОВИЧУ

1. Тема роботи: «Розробка інформаційної системи сповіщення для покращення роботи військкомату з використанням об'єктно-орієнтовних мов програмування»

Керівник роботи Негоденко Олена Василівна, завідувач кафедри ПІЗ

затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи «01» червня 2021 року

Вхідні дані до роботи:

Розробка інформаційних систем сповіщення.

Алгоритм системи сповіщення військкомату.

Науково-технічна література, пов'язана з розробкою застосунків інформаційних систем призначених для отримання сповіщень.

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

4.1 Аналіз вимог до інформаційної системи.

4.2 Аналіз та порівняння існуючих рішень для надсилання сповіщень від військкомату.

4.3 Дослідження програмних засобів для розробки інформаційної системи.

4.3. Перелік демонстраційного матеріалу (назва основних слайдів)

1. _____

2. _____

3. _____

4. _____ 5.

Дата видачі завдання «19» квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04-23.04	Виконано
2	Аналіз вимог до інформаційної системи.	23.04-26.04	Виконано
3	Аналіз існуючих систем для надсилання сповіщень військкоматом.	26.04-28.04	Виконано
4	Дослідження програмних засобів	28.04-30.04	Виконано
5	Моделювання об'єкту проектування	01.05-05.05	Виконано
6	Розробка функціоналу застосунку	05.05-15.05	Виконано
7	Вступ, висновки, реферат	15.05-17.05	Виконано
8	Розробка презентації застосунку	18.05	Виконано

Студент

Керівник роботи

РЕФЕРАТ

Текстова частина бакалаврської роботи 51с., 36 рис., 19 джерел

Ключові слова: Інформаційна система, чат-бот, месенджер, Telegram, IDE, система контролю версій, вимоги

Об'єкт дослідження – процес передачі інформації для покращення роботи військкомату.

Предмет дослідження – інформаційна система для покращення процесу надсилання сповіщень військкоматом.

Мета роботи – підвищення ефективності сповіщення шляхом розробки інформаційної системи для покращення роботи військкомату з використанням об'єктно орієнтованих мов програмування.

Методи дослідження – методи надсилання сповіщень військкоматом, методи структурного аналізу і проектування, методи розробки програмного забезпечення.

Наукова новизна даної роботи полягає в наступному:

1. Встановлено, що використання об'єктно-орієнтованих мов програмування та відповідних середовищ розробки - це вдале рішення для досягнення поставленої мети.
2. Показано, що мова об'єктно орієнтовані мови зручні у використанні та надають можливість ефективної розробки застосунку.
3. На основі результатів виконаних досліджень розроблено інформаційну систему надсилання сповіщень військкоматом.

Впровадження розробленої інформаційної системи допоможе оптимізувати роботу військкомату за рахунок створення новітніх процесів для надсилання сповіщень. Використання інформаційної системи допоможе своєчасно та ефективно отримувати сповіщення від військкомату за рахунок використання месенджера «Telegram»

В роботі виконано аналіз існуючих систем сповіщення військкоматом, виявлено їх суттєві недоліки та переваги. В результаті цього було визначено основні потреби користувачів.

Галузь використання – надсилання інформаційних повідомлень. Щодо кінцевого використання, тобто отриманню сповіщень жодних обмежень не існує.

ЗМІСТ

ВСТУП.....	9
1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ НАДСИЛАННЯ СПОВІЩЕНЬ ВІЙСЬККОМАТОМ.....	11
1.1. Огляд та аналіз існуючих месенджерів	11
1.2. Аналіз вимог до інформаційної системи	13
2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ НА ОСНОВІ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ	17
2.1. Вибір мови програмування	17
2.2. Вибір середовища розробки.....	20
2.3. Інтерфейс та основні компоненти обраного середовища розробки	28
2.4. Аналіз процесу створення чат-боту в месенджері «Telegram».	34
2.5. Використання системи контролю версій Git та Github.....	37
3 ПРОЕКТУВАННЯ МАКЕТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ	41
3.1. Огляд програмних засобів для створення макетів.....	41
3.2. Створення макетів інформаційної системи.....	46
4 ІНСТРУКЦІЇ ЩОДО ВИКОРИСТАННЯ СИСТЕМИ	49
4.1. Інструкція менеджера системи	49
4.2. Інструкція кінцевого користувача.....	57
4.3. Тестування інформаційної системи	58
ВИСНОВКИ.....	60
СПИСОК ЛІТЕРАТУРИ.....	62
Додаток А.....	64
Додаток Б.....	66
Додаток В	70

ВСТУП

Актуальність.

Об'єкт дослідження – процес передачі інформації для покращення роботи військомату

Предмет дослідження – інформаційна система для покращення процесу надсилання сповіщень військоматом.

Мета роботи – підвищення ефективності сповіщення шляхом розробки інформаційної системи для покращення роботи військомату з використанням об'єктно орієнтованих мов програмування.

Методи дослідження – методи надсилання сповіщень військоматом, методи структурного аналізу і проектування, методи розробки програмного забезпечення.

У відповідності до поставленої мети визначено основні завдання дослідження:

- проаналізувати поточний стан задач;
- дослідити можливості використання засобів інтеграції месенджера Telegram з мовою програмування PHP;
- встановити відповідність вимогам об'єктно-орієнтованого аналізу для розробки інформаційної системи;
- Розробити структуру бази даних системи.
- Розробити програмне забезпечення для створення інформаційної системи та її інтеграції з месенджером Telegram;

Наукова новизна даної роботи полягає в наступному:

- Розроблено об'єктно орієнтовану модель інформаційної системи;
- Встановлено що використання мови програмування PHP повністю задовольняють відповідні вимоги;
- Показано що архітектура додатку забезпечує необхідні функціональні можливості системи;

Практична значущість полягає в тому, що результати дослідження можуть бути використані у військоматах для додаткового спроби інформування призовників.

Результати дослідження бакалаврської роботи апроксимовані на наступних конференціях:

- Застосування програмного забезпечення в інфокомунікаційних технологіях

1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ НАДСИЛАННЯ СПОВІЩЕНЬ ВІЙСЬККОМАТОМ

1.1. Огляд та аналіз існуючих месенджерів

На сьогоднішній день, у кожного з нас на мобільному телефоні встановлено один, або навіть декілька месенджерів.

Месенджер - додаток що призначений для миттєвого обміну текстовими повідомленнями, фото та відео, документами з іншими користувачами месенджеру. Як правило, месенджери працюють використовують службу обміну миттєвими повідомленнями, що підтримує зв'язок в реальному часі, між користувачами через мережу інтернет.

До найвідоміших месенджерів можна віднести наступні:

- Viber;
- WhatsApp;
- Facebook Messenger;
- Telegram.

Таблиця 1.1 - Огляд існуючих месенджерів[1]:

	Telegram	Viber	WhatsApp
Компанія	Telegram FZ LLC	Rakuten, Inc.	Facebook
Тип	Некомерційний	Комерційний	Комерційний
Ідея	Свобода безпеки та приватності спілкування доступна для кожного	"Максимально захищений месенджер." Резервні копії не зашифровані.	"Конфіденційність та безпека – частина нашої ДНК." Резервні копії не зашифровані, виявлені вразливості.

Продовження таблиці 1.1.

Документація	Відкрита та деталізована	Відкрита та деталізована (розробка нових додатків неможлива)	Практично відсутня
Кількість Активних Користувачів	500 мільйонів	260 мільйонів	2000 мільйонів
Кількість Пристроїв	Необмежена	Один первинний та декілька вторинних пристроїв	Два
Підтримувані ОС	Android, iOS, Windows, macOS, Linux, Wear OS, watchOS	Android, iOS, Windows, macOS, Linux, Wear OS, watchOS	Android, iOS, Windows, macOS
Текстові повідомлення	З підтримкою комбінованого форматування (жирний, моноширинний, курсивний, підкреслений, закреслений шрифт)	Без підтримки форматування	З підтримкою форматування (жирний, моноширинний, курсивний, закреслений шрифт)
Аудіо повідомлення	Підтримується	Не підтримується (виключно як файли)	Підтримується
Відео повідомлення	Підтримуються	Підтримуються	Підтримуються (до 0,016 ГБ)
Надсилання файлів	До 2 ГБ в необмежених кількостях	До 0,2 ГБ	Медіа до 0,016 ГБ, файли до 0,1 ГБ

1.2. Аналіз вимог до інформаційної системи

Інформаційна система - сукупність інформаційних, технічних, програмних та технічних засобів, які необхідні для автоматизованої обробки певної інформації.

Тобто, вимоги до інформаційної системи, є складаються з вимог до апаратного та програмного забезпечення. Вимоги до інформаційної системи, є ідентичними вимогам до програмного забезпечення.

Вимоги до програмного забезпечення - це набір вимог, які визначають особливість, якість та функції програмного забезпечення. Варто відзначити що вимоги визначаються в рамках окремого етапу розробки програмного забезпечення "Аналіз вимог"[2].

Аналіз вимог полягає у визначенні потреб та умов, які запропоновані до уже існуючого ПЗ, або нового ПЗ (тобто те, яке потрібно розробити).

Типи вимог:

1. Архітектурні вимоги.
2. Функціональні вимоги.
3. Нефункціональні вимоги.

Архітектурні вимоги - це вимоги, які визначають необхідний стек технологій згідно з яким повинно бути розроблене програмне забезпечення. Також включають в себе вимоги щодо надійності та захищеності програмного забезпечення.

Функціональні вимоги - це вимоги, які визначають необхідний функціонал, поведінку, правила обробки даних для інформаційної системи, що розробляється.

Нефункціональні вимоги - це вимоги, які визначають критерії для оцінки якості роботи програмного забезпечення, наприклад вимоги до інтерфейсу, до безпеки даних та інші[17].

Методи визначення вимог:

1. Спілкування з майбутнім користувачем.
2. Brainstorm (Мозковий штурм).
3. Аналіз існуючої документації та чинного законодавства.

4. Аналіз існуючих бізнес-процесів в межах підприємства[18].

Спілкування з майбутнім користувачем - процес, під час якого визначаються основні вимоги користувача до програмного забезпечення. може проходити в якості:

1. Бесіди.
2. Інтерв'ю.
3. Анкетування.
4. Спостереження.

Brainstorm(Мозковий штурм) - метод, основною ідеєю якого є процес стимулювання творчого мислення. Відбувається як самотійно(1 особа) так і колективно(як правило, до 12 осіб). Метою є висловлення максимальної кількості ідей, витрачаючи на кожну до 5 хвилин.

Аналіз існуючої документації та чинного законодавства - процес, під час якого проводиться дослідження уже існуючої документації та законодавства. З результатів дослідження формулюються ряд певних вимог до програмного забезпечення[3].

Аналіз існуючих бізнес-процесів всередині компанії - процес, під час якого, проводиться дослідження та вивчення існуючих бізнес процесів в межах підприємства. В результаті дослідження визначаються пріоритетні бізнес-процеси, та процеси які можна вдосконалити в новому програмному забезпеченні.

Під час формування вимог для інформаційної системи надсилання сповіщень військкоматом, було обрано наступні методи аналізу вимог:

1. Спілкування з майбутнім користувачем.
2. Brainstorm (Мозковий штурм).

В результаті яких, було сформовано наступні функціональні вимоги до інформаційної системи надсилання сповіщень військкоматом:

1. Можливість надсилання шаблонних повідомлень - тобто необхідно реалізувати можливість, відправки одного шаблонного повідомлення на певне коло отримувачів в заданий час.

2. Можливість надсилання інформаційних повідомлень - тобто необхідно реалізувати можливість, відправки одного інформаційного повідомлення на певно коло отримувачів, в момент підтвердження відправки повідомлення.
3. Можливість створення доступів в інформаційної системи - тобто адміністратор системи, повинен мати можливість додавати та видаляти доступи в інформаційну систему. Це дозволяє керувати керувати адміністративними доступами до інформаційної системи в режимі реального часу.
4. Одночасність доступу до інформаційної системи - тобто, в режимі реального часу, де кілька адміністраторів мають змогу паралельно працювати в інформаційній системі.
5. Історія надісланих повідомлень - всі надіслані повідомлення повинні бути зареєстровані в інформаційній системі, та бути доступними для перегляду їх в любий момент часу.
6. Можливість керування статичною інформацією - такою як "Контакти".

Також, в результаті аналізу вимог до інформаційної системи, було сформовано наступні нефункціональні вимоги:

1. Захист від SQL ін'єкцій.
2. Шифрування паролів в БД.
3. Наявність HTTPS протоколу та SSL сертифікату.

SQL ін'єкція - один із поширених способів способів злому сайтів та програм, що працюють з базами даних. Основа методу полягає в провадженні в запит довільного SQL коду. Що в свою чергу, надає атакуючому можливість виконати запит до бази даних. Наслідками успішної SQL ін'єкції може бути:

1. Викрадення персональних даних користувачів.
2. Модифікація облікових записів як користувачів, так і адміністраторів інформаційної системи.
3. Видалення таблиць, полів, даних.

Для того, щоб убезпечити систему від SQL ін'єкцій необхідно дотримуватись наступних рекомендацій:

1. Приведення значень до цілочисельного типу даних. Оскільки SQL запити, часто виконуються з числами наприклад як id запису, то їх, необхідно приводити до цілочисельного типу даних. В результаті приведення, якщо зловмисник підставив замість числа вираз, результатом виразу буде нуль, що і дозволить уникнення виконання SQL ін'єкції.
2. Екранування значень та виразів SQL запиту. Основа метода полягає в тому, що всі підозрілі значення у виразі необхідно за допомогою спеціальних функцій екранувати за допомогою символу “\”.

Шифрування паролів в БД. Оскільки існують ситуації коли конфіденційні дані користувачів та адміністраторів системи із бази даних потрапляють до зловмисників, гарною практикою є шифрування інформації в базі даних. Як правило завжди зашифровується пароль користувача. Оскільки маючи лише зашифрований пароль - зловмисник не може його використати для взлому аккаунта.

HTTPS - розширення протоколу HTTP, що підтримує шифрування за допомогою TLS та SSL сертифікатів.

HTTP - прикладний протокол передачі даних в мережі. На даний час його використовують для отримання інформації з веб-сайтів.

Відмінності HTTPS та HTTP:

1. HTTPS це всього лиш доповнений шифруванням протокол HTTP.
2. Дані що передаються по протоколу HTTPS є надійно зашифрованими, на відміну від протоколу HTTP.

SSL - криптографічний протокол, що забезпечує встановлення безпечного з'єднання між клієнтом та сервером. Завдяки протоколу SSL всі дані, обмін якими відбувається за допомогою протоколу TCP/IP є надійно захищеними.

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ НА ОСНОВІ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Вибір мови програмування

На сьогоднішній день існує безліч мов програмування з різним функціональним призначенням та для рішення різних задач. Наприклад мова програмування Swift що була представлена на щорічній конференції розробників WWDC 2014 чудово підходить для розробки програм та додатків для девайсів на платформі iOS, MacOS, WatchOS, IpadOs. Аналогічно мові програмування Swift для девайсів з операційною системою Android існує мова програмування під назвою Go, що була представлена в листопаді 2009 року. Про те, ці мови програмування не є заточеними під конкретні задачі, а є універсальними. Так само і з іншими мовами програмування, наприклад мова програмування Java є суворо типізованою та чудово підходить для реалізації принципів ООП. Аналогічною мові Java є мова програмування C#[5], що була розроблена корпорації Microsoft як аналог мові програмування Java, та згодом визначена як флагманська мова програмування корпорації Microsoft. Існують і інші мови програмування, наприклад Prolog є мовою логічного програмування. Основами мови Prolog є математична логіка та логіка першого порядку, що робить цю мову зручною для вирішення специфічних задач. Також варто відзначити мову програмування Python, яка на сьогоднішній день є однією з найактуальніших для створення штучного інтелекту[6].

Таблиця 2.1 - Порівняння мов програмування:

Мова	Python	Prolog	PHP	Java	C#
Передбачу ване використання	Додатки, Web, Штучний інтелект	Додатки, Штучний інтелект	Серверна частина, Web, Додатки	Серверна частина, Web, Додатки	Серверна частина, Web, Додатки, програмування роботів

Продовження таблиці 2.1.

Мова	Python	Prolog	PHP	Java	C#
Імперативна	Так	Ні	Так	Так	Так
Об'єктно орієнтовна	Так	Ні	Так	Так	Так
Функційна	Так	Ні	Ні	Ні	Так
Процедурна	Ні	Ні	Ні	Ні	Так
Подійно орієнтована	Ні	Ні	Ні	Ні	Так

Отже, роблячи висновки з таблиці 2.1. Порівняння мов програмування можна відзначити що найбільш універсальною мовою програмування є C#. Проте для створення інформаційної системи для надсилання сповіщень військкоматом було обрано мову програмування PHP.

PHP - мова програмування що призначення для створення HTML сторінок на стороні веб-сервера. На відміну від мови програмування JavaScript клієнтська частина (тобто веб-браузер) отримує готовий HTML код сторінки.

Особливості PHP[7]:

1. Наявність вбудованих інтерфейсів для роботи з базами даних. За замовчуванням в PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, SQLite, mSQL, Oracle, InterBase.
2. Нетрадиційність. Оскільки PHP має запозичений синтаксис з мови C та Perl, то програмісти в тому числі, ті що лише починають вивчати програмування в цілому - дуже легко звикають до багатьох конструкції мови PHP.
3. OpenSource та безкоштовність. Саме завдяки цим стратегіям, мова PHP досить швидко стала популярною у колі програмістів. З часом з'явилося ком'юніті, яке охоче допоможе з вирішенням навіть найскладніших питань.

4. Ефективність. PHP є інтерпретованою мовою програмування, що в свою чергу дозволяє здійснювати обробку сценаріїв з достатньо великою швидкістю.

Основи синтаксису PHP:

1. Змінні. Оголошення змінних відбувається після зарезервованого символу "\$". Приклади оголошення змінних:
 - a. \$age;
 - b. \$favorite_color;
 - c. \$name2
2. Операція присвоювання. Для присвоєння значення змінній використовується символ "=". Приклад операції присвоєння:
 - a. \$favorite_color = "green";
 - b. \$favorite_number = 42;
3. Оператори порівняння - "==", "!=", ">", "<", ">=" "<=". Приклади використання:
 - a. "\$a == \$b" - \$a рівно \$b;
 - b. "\$a != \$b" - \$a нерівно \$b;
 - c. "\$a > \$b" - \$a ільше \$b;
 - d. "\$a < \$b" - \$a енше \$b;
 - e. "\$a >= \$b" - \$a більше або рівно \$b;
 - f. "\$a <= \$b" - \$a менше або рівно \$b.
4. Оператор ділення з залишком - "%" Використовується тільки для чисел. Приклади використання:
 - a. 5 % 2 == 1;
 - b. 10 % 5 ==0;
5. Відокремлення інструкцій. Для відокремлення інструкцій використовується символ ";" (точка з крапкою).
6. Коментарі. В синтаксисі PHP розрізняють наступні коментарі:
 - a. Однорядкові - позначаються символом "//", або символом "#";

- в. Багаторядкові - починаються з комбінації символів “/*” та закінчуються комбінацією символів “*/”;
7. Вставка коду в шаблон HTML сторінки. Для вставки коду в HTML необхідно скористатись початковим тегом “<?php” та кінцевим тегом “?>”, поміж яких розмістити код PHP.

Приклад найпростішої програми на мові PHP:

```
1. <?php
2.     echo 'Hello, world!';
3. ?>
```

Приклад найпростішої програми з використанням шаблону HTML:

```
1. <head>
2.     <title>Тестуємо PHP</title>
3. </head>
4. <body>
5.     <?php echo 'Hello, world!'; ?>
6. </body>
7. </html>
```

2.2. Вибір середовища розробки

Інтегроване середовище розробки - багатофункціональний програмний додаток, що забезпечує всі необхідні потреби програмістів для розробки програмного забезпечення. Як правило, середовище розробки складається з:

1. Редактора коду.
2. Компілятор або інтерпретатор.
3. Засоби автоматизації збирання проекту.
4. Відладчик коду.

Редактор коду - окрема комп'ютерна програма, або компонент програмного додатку, який призначено для створення та редагування програмного коду, рис 2.1. Зазвичай редактори коду містять додатковий функціонал, який призначений спростити роботу програмісту, а саме: підсвітка синтаксису, підсвітка програмного коду, сортування рядків, конвертацію кодування символів, та інше.

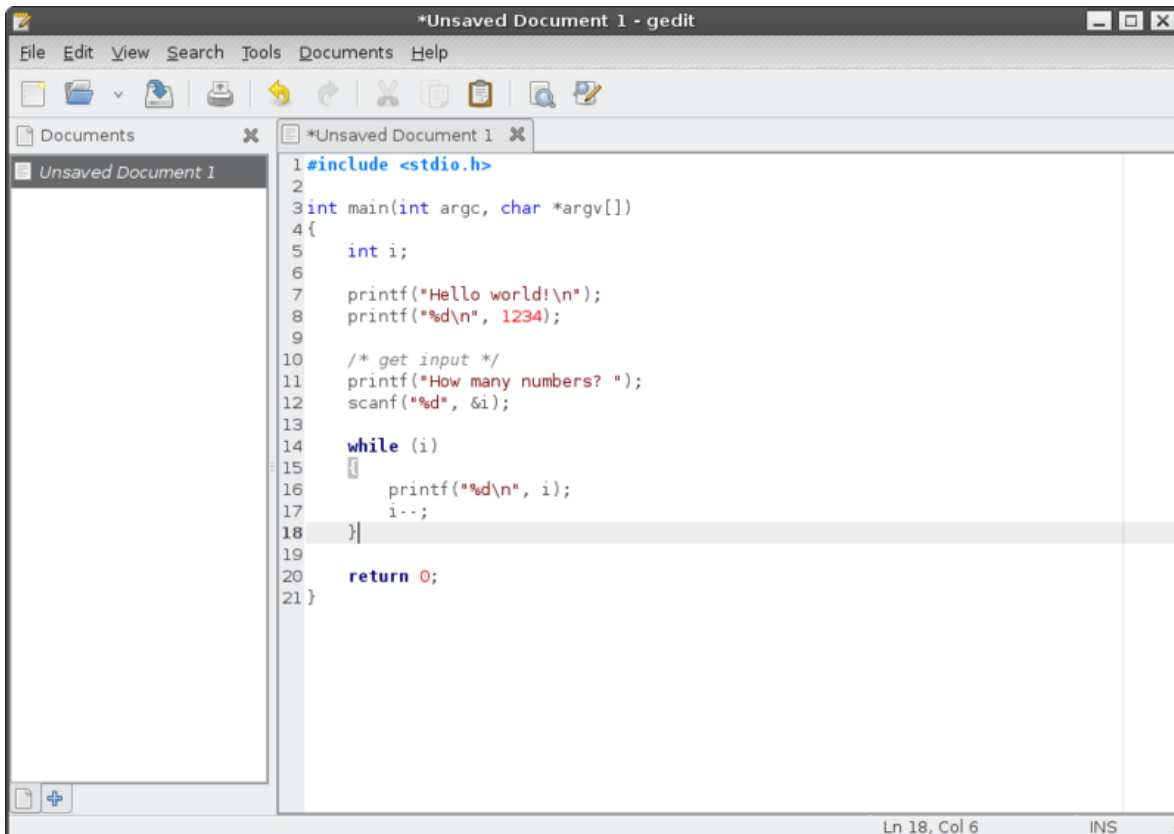


Рисунок 2.1. - Вікно текстового редактора gedit.

Компілятор - програма, або частина програмного засобу, що призначена для переведення програмного коду у двійковий код, або інший низькорівневий програмний код, що буде зрозумілий комп'ютеру.

Компіляція - процес збирання проекту, що складається з наступних етапів:

1. Переведення всіх блоків програми з мови високого рівня, на мову низького рівня, або безпосередньо в двійковий код.
2. Збирання проекту на виконуваному комп'ютері.

Інтерпретатор - один з видів транслятора, що займається здійсненням пооператорної обробки коду в машинний код, та безпосереднє виконання програми. Інтерпретатори бувають простими, та компілюючого типу. Характерною відмінністю між ними є те, що інтерпретатор компілюючого типу переводить програмний код у проміжне представлення наприклад як байт-код, або р-код, в результаті отримуючи проміжний код. В той же час простий інтерпретатор, аналізує та виконує кожен оператор покомандно, що забезпечує кращу реакцію на виконання команди. До недоліків простого інтерпретатора можна віднести алгоритм виявлення помилок, оскільки виявлені вони будуть, лише в момент виконання тієї чи іншої команди.

Відладчик коду - програма, або частина програмного засобу, що призначена для автоматизації процесу відладки коду, пошуку помилок в кодї, SQL-запитах та інших видах коду. Для програмування мовою PHP доступні наступні інтегровані середовища програмування:

1. PhpStorm.
2. PhpStorm 8.
3. NetBeans.
4. Komodo IDE.
5. ZendStudio.
6. Kite.

PhpStorm - комерційне інтегроване середовище розробки на мові програмування PHP, з інтелектуальним редактором коду.

Основні можливості PhpStorm:

1. Інтелектуальний редактор коду з підсвіткою синтаксиса, автозаповненням.
2. Підтримка PHP 7.0, 5.6, 5.5, 5.4.
3. Виявлення програмного коду що повторюється у декількох місцях.
4. Вбудований REST клієнт, наявність SSH консолі.
5. Підтримка HTML, JavaScript та CSS.
6. Інтеграція з системами використання версій програмного продукту.

7. Віддалений запуск програмних додатків та автоматична синхронізація з FTP сервером.
8. Широкий набір вбудованих інструментів для роботи з базами даних, наявність вбудованого SQL редактору[8].

PhpDesigner 8 - комерційний редактор коду та IDE, рис. 2.2., що призначені для створення сучасних web додатків з використанням PHP, HTML, JavaScript та CSS.

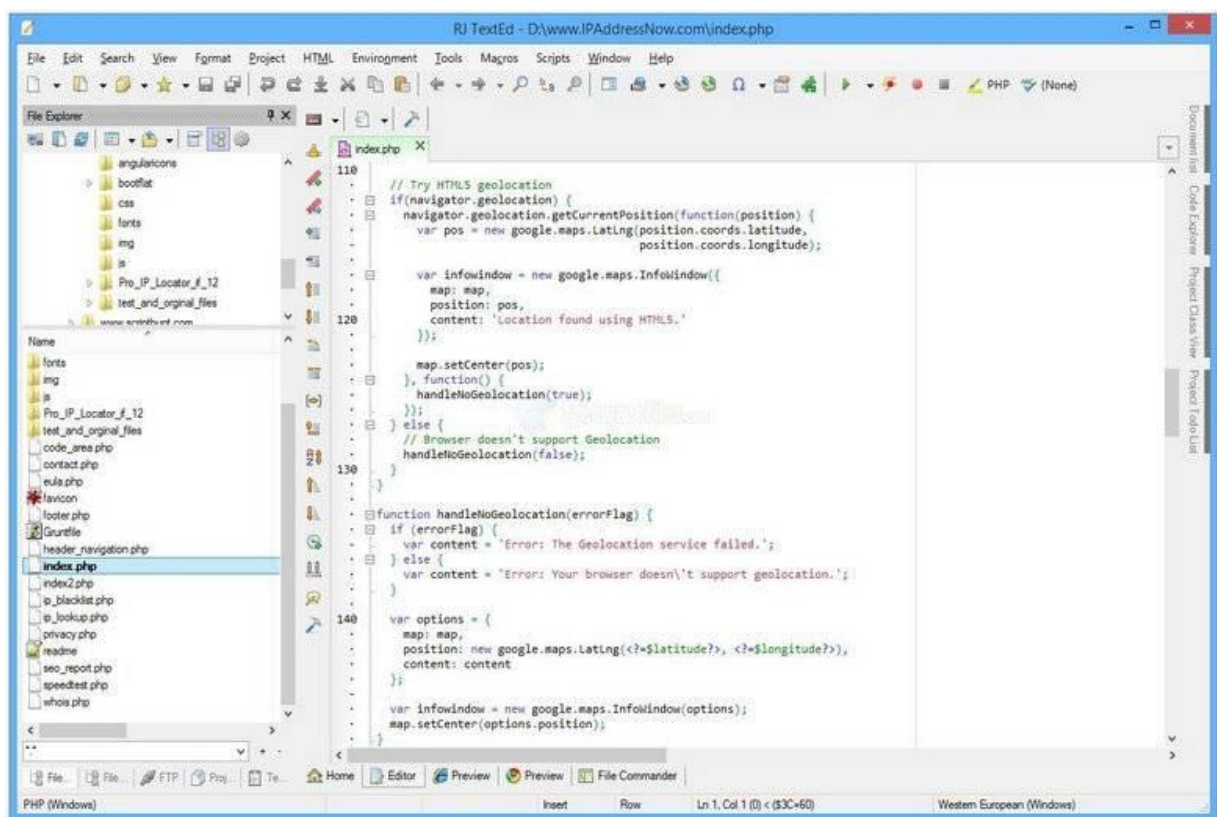


Рисунок 2.2. - редактор коду PhpDesigner 8.

Ключові можливості:

1. Широка підтримка PHP фреймворків, таких як: ZENDE, YII, YII2, Codeigniter, Symphony, CakePHP, Laravel.
2. Автозбереження проекту на сервері FTP.
3. Інтеграція з системою контролю версій Git.
4. Перевірка в режимі реального часу синтаксису, підсвітка помилок.

5. Підтримка та інтеграція TortoiseSVN.
6. Вбудований менеджер задач та помилок.
7. Редактор баз даних (інтеграція з PhpMyAdmin).

NetBeans - безкоштовна IDE призначена для програмування на мові Java, рис. 2.3. додатково, NetBeans IDE має розширення для програмування на наступних мовах: C, C++, PHP, JavaScript, HTML5.

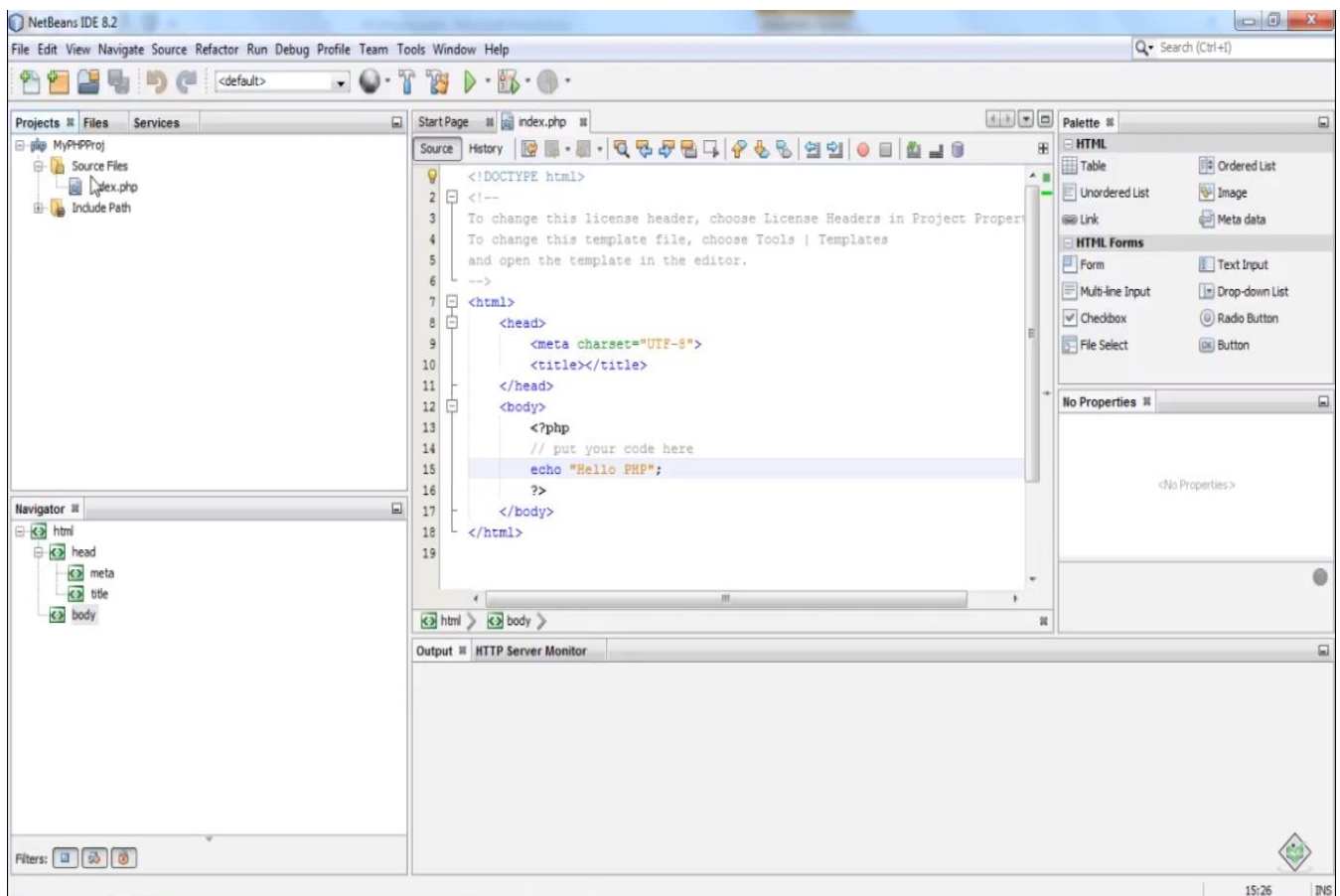


Рисунок 2.3. - середовище розробки NetBeans IDE.

Ключові можливості для мови програмування PHP:

1. Підсвітка синтаксису, автодоповнення конструкції коду, підсвітка синтаксичних помилок.
2. PHP дебагінг коду за допомогою розширення xdebug.
3. Модульне тестування PHP за допомогою вбудованого фреймворку PHPUnit та Selenium.

4. Підтримка фреймворку Symfony починаючи з версії 6.8.
5. Підтримка фреймворку Zend починаючи з версії 6.9.
6. Підтримка фреймворку Yii починаючи з версії 7.3.

Komodo IDE - інтегроване середовище розробки для динамічних мов програмування. Komodo IDE підтримує наступні мови програмування: Perl, Ruby, PHP, HTML, Python, CSS, рис 2.4.

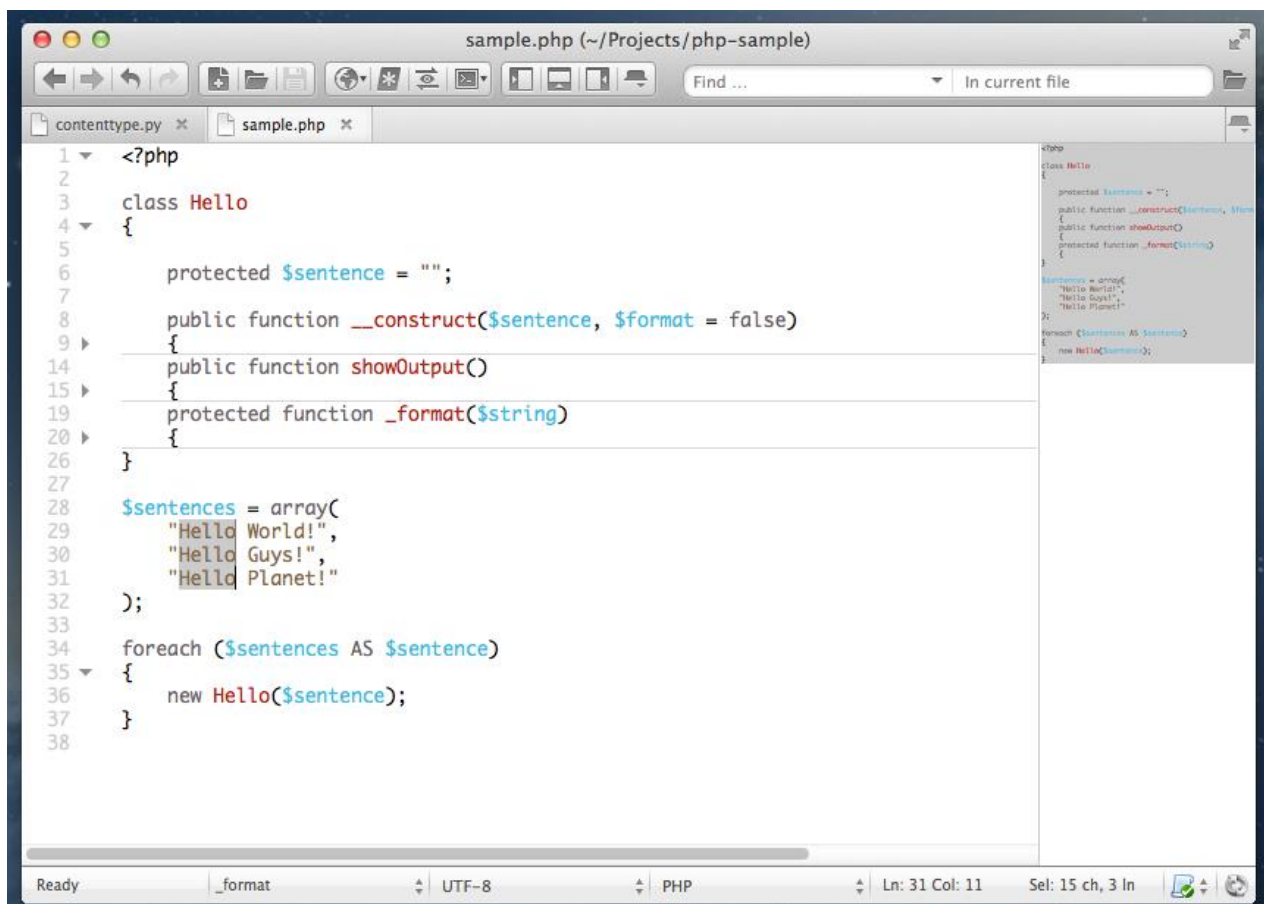


Рисунок 2.4. - інтегроване середовище розробки Komodo IDE.

Ключові можливості:

1. Редактор коду з автозаміною, автозаповненням, підсвіткою синтаксису та помилок.
2. Підтримка систем контролю версій, таких як: Git, Mercurial, Subversion, CVS.

3. Вбудований HTTP Inspector, що дозволяє візуально протестувати регулярні вирази.
4. Вбудований магазин розширень та доповнень для IDE.
5. Віддалене редагування коду з використанням одного з наступних протоколів: FTP, FTPS, SFTP, SCP.

ZendStudio - комерційне інтегроване середовище розробки для мови програмування PHP, рис. 2.5., що було розроблене компанією ZendTechnologies та засноване на плагіні PHP Development Tools.

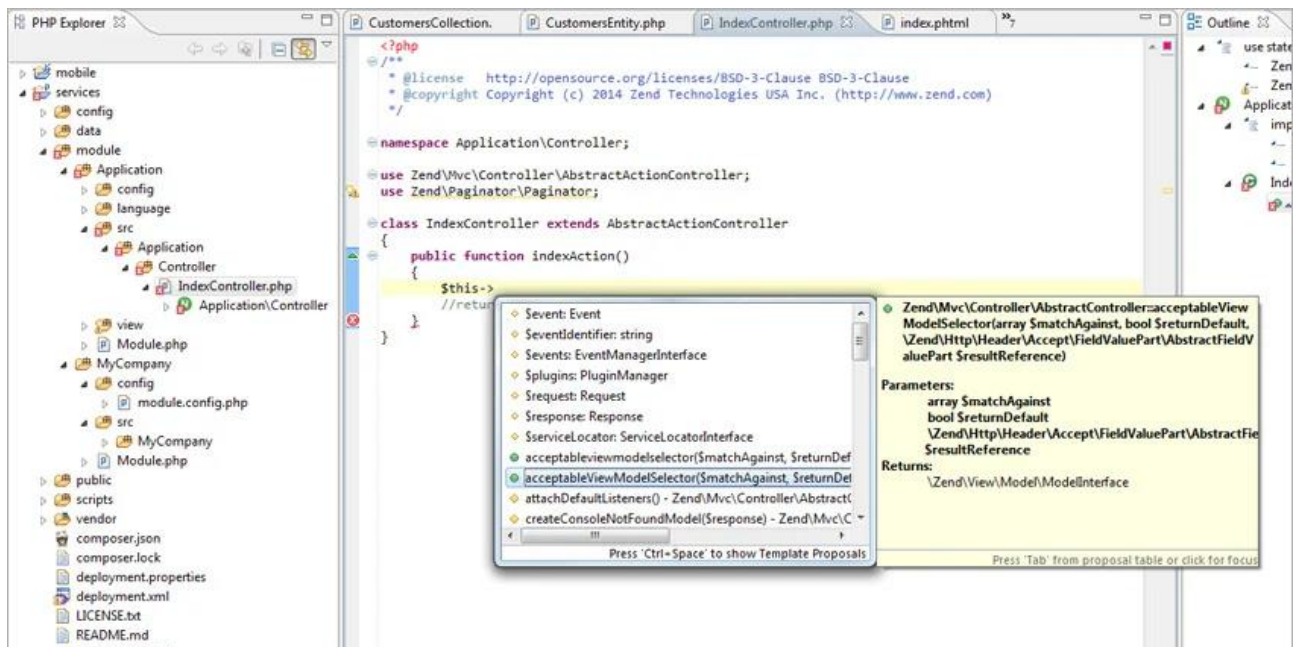


Рисунок 2.5. - інтегроване середовище розробки
ZendStudio.

Ключові можливості:

1. Аналіз та авто виправлення програмного коду.
2. Підсвітка синтаксису, автодоповнення, підказки.
3. Інтеграція з Zend Framework.
4. Підтримка мови програмування PHP починаючи з версії 4.0.
5. Вбудовані засоби модульного тестування PHPUnit.
6. Підтримка мов розмітки: CSS та HTML.

7. Підтримка віддалених серверів за допомогою протоколів FTP, SFTP, SSH.
8. Підтримка перегляду наступних систем управління БД: MySQL, Microsoft SQL Server, Oracle, PostgreSQL та SQLite.

Kite - редактор коду, заснований на технологіях штучного інтелекту, що допомагає оптимізувати процес написання програмного коду, за рахунок інтелектуальних підказок та автодоповнень рис. 2.6. Kite підтримує наступні мови програмування: Java, JavaScript, Go, C Based, Scala, TypeScript, Kotlin, PHP, Ruby та в преміум версії редактора Python.

```

fly_with_kite.py
1 import json
2 json.dumps(l)

json.dumps(obj, skipkeys=False,
ensure_ascii=True,
check_circular=True, allow_nan=True,
cls=None, indent=None,
separators=None, encoding="utf-8",
default=None, sort_keys=False, **kws)

HOW OTHERS USED THIS
dumps(obj)
dumps(obj, indent=4)
dumps(obj, indent=4, sort_keys=True if
pretty else False)
dumps(obj, cls=json.JSONEncoder if
self.compact else DateJSONEncoder)
dumps(obj, default=lambda o: o.__dict__)

Docs
fly_with_kite.py* 2:12 LF UTF-8 Python GitHub Git (0) 1 update

```

Рисунок 2.6. - редактор коду Kite.

Ключові можливості:

1. Автодоповнення коду на основі штучного інтелекту.
2. Підсвітка синтаксису, авто виправлення помилок.
3. Можливість інтеграції в IDE інших компаній.

4. Підтримка PHP 7.0, 5.6, 5.5, 5.4.
5. Виявлення програмного коду що дублюється у кодї програмного продукту.

2.3. Інтерфейс та основні компоненти обранного середовища розробки

Провівши дослідження серед найпопулярніших середовищ розробки та редакторів коду, було прийнято рішення щодо використання в якості середовища розробки IDE PhpStorm від компанії розробників JetBrains. Оскільки саме ця IDE має зрозумілий для кінцевого користувача (програміста) інтерфейс, та інші переваги, що перелічені в розділі 2.2. Єдиним недоліком цієї IDE є лише наявність платної версії. Проте студент вищого навчально закладу, завжди може отримати ліцензію на один рік використання IDE в навчальних цілях, після підтвердження.

Навігаційний бар IDE PhpStorm містить наступні вкладки, рис 2.7.[8]:

1. File.
2. Edit.
3. View.
4. Navigate.
5. Code.
6. Refactor.
7. Run.
8. Tools.
9. VCS.
10. Window.
11. Help

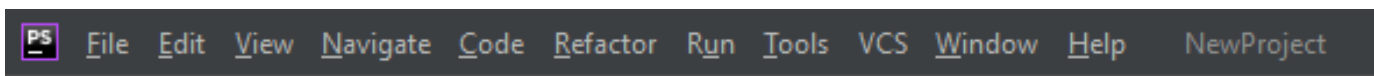


Рисунок 2.7. - Навігаційний бар IDE PhpStorm.

Вкладка “File”, рис.2.8. - призначена для виконання операцій над конфігураційними файлами проекту (таких як: додавання до проекту нових файлів, збереження змін в проекті, перейменування проекту, керування налаштуваннями проекту), створення нового проекту, скидання закешованих файлів, завантаження та збереження змін на віддаленого диску, експорт файлів у формат HTML, керування друком програмного коду та керування налаштуваннями самої IDE.

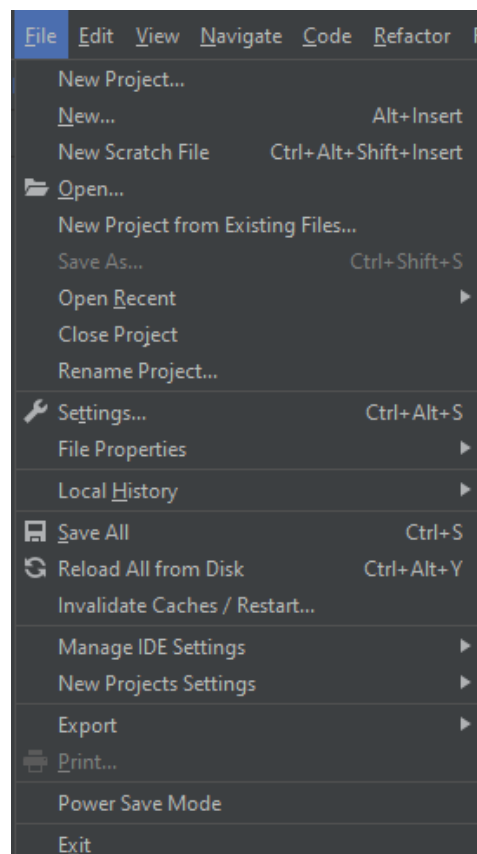


Рисунок 2.8. - перелік доступних опцій у вкладці “File”.

Вкладка “Edit”- призначена для виконання простих операцій з програмним кодом у текстовому редакторі. Прості операції включають в себе: вирізати/скопіювати/видалити/вставити програмний код у виділений відрізок, знайти/знайти та замінити вказаний програмний код, здублювати/відсортувати/реверсувати певний відрізок програмного коду, та інші подібні операції.

Вкладка “View”, рис 2.9. - призначена для перегляду доступних інструментів, для налаштування зовнішнього вигляду IDE PhpStorm для більш зручного використання, перегляду нещодавно відредагованих файлів з програмним кодом, та позначенням змін в них, які нещодавно були здійснені, перегляду останніх директорій файлової системи та швидкої зміни теми середовища розробки (зі світлої на темну, та навпаки).

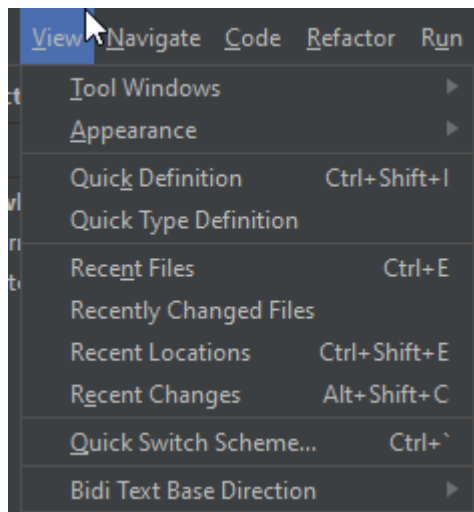


Рисунок 2.9. - перелік доступних опцій у вкладці “View”.

Вкладка “Navigate” - призначена для швидкої навігації в програмному коді. Наприклад, під час рефакторингу коду, можуть виникнути ситуації коли помилки в програмному будуть з’являться одна за одною, для таких випадків доречно використовувати доступну опції “Next Highlight Error”/“Previous Highlight Error”, що буде прокручувати програмний код до наступного/попереднього місця виникнення помилки. Під час процесу дебагінга програмного коду, зручно використовувати опцію “Next Emmet Edit Point”/“Previous Emmet Edit Point”, що буде прокручувати програмний код до наступної/попередньої мітки редагування. До інших зручних опцій, розташованих у вкладці “Navigate” можна віднести: перехід до наступного/попереднього методу в програмному коді, огляд місця розташування файла з програмним кодом, відображення структури файлу та інше.

Вкладка “Code” - призначена для спрощення процесу написання програмного коду, наприклад, зручними для використання є опції “Override Methods” та “Implement Methods”, опція коментування програмного коду (однострокова та багатострокова), авто табуляція програмного коду, що значно покращує його читаємість, опції переміщення виділеного програмного коду (одностроково та багатостроково) вгору/вниз, для дотримання правильної послідовності виконання алгоритму.

Вкладка “Refactor”, рис. 2.10. - призначена для оптимізації процесу рефакторингу програмного коду, основними опціями, окрім запуску процесу рефакторингу є: перейменування файлу з програмним кодом, безпечне видалення файлу з програмним кодом, копіювання та переміщення файлу з програмним кодом до іншої директорії, та інші операції.

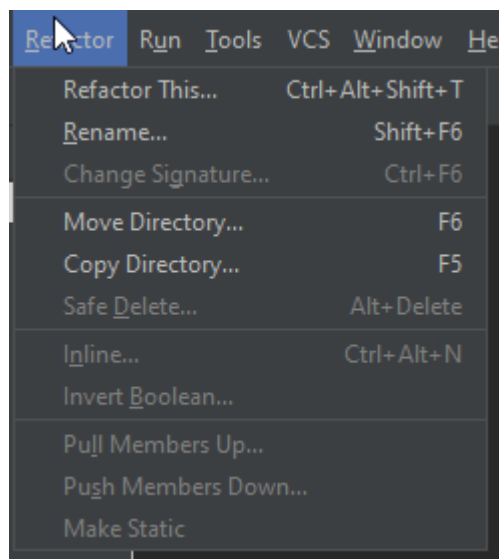


Рисунок 2.10. - перелік доступних опцій у вкладці “Refactor”.

Вкладка “Run” - призначена для збирання, запуску проекту, налаштування його конфігурацій, додавання нових точок зупинки для виконання процесу дебагінгу, операції покрокового виконання програмних інструкцій (крок із

заходом всередину операцій, операції зупинки та продовження виконання програмних інструкцій), та інші операції.

Вкладка “Tools” - призначена для запуску вбудованих в IDE інструментів, таких як: контроль виконання задач, відкриття вбудованої консолі IDE Scripting Console, виконання операції із текстом XML розмітки, операції з “деплоєм” на сервері (такі як: відправлення останніх змін на сервер, завантаження останніх змін з серверу, синхронізація змін з сервером, налаштування конфігурацій, та налаштування графіка відправлення останніх змін на сервер), та інших (“V8 Profiling”, “HTTP Client”, “DBGp Proxy”, “XDebug” та “Composer”).

Вкладка “VCS” - розшифровується як “Version Control System”, тобто система керування версіями, рис. 2.11. Основними опціями є: включення та налаштування системи контролю версій, створення репозиторію на GitHub та Mercurial, розповсюдження проекту на GitHub.

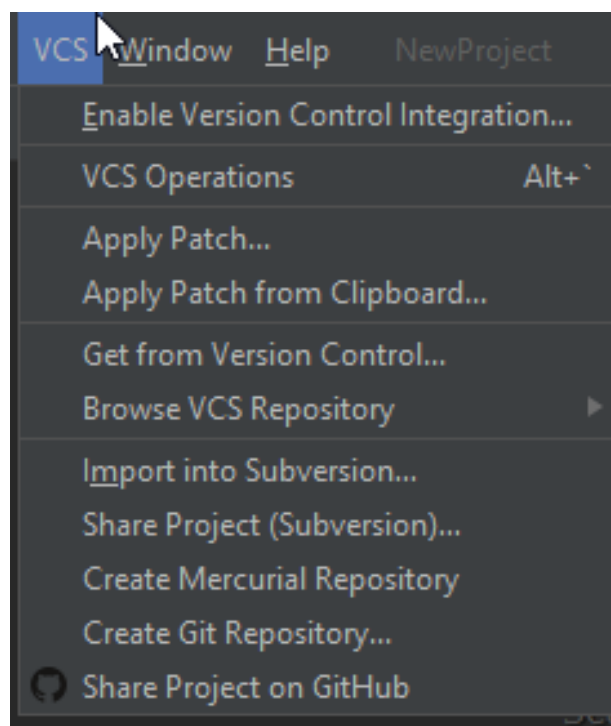


Рисунок 2.11. - перелік доступних опцій у вкладці “VCS”.

Вкладка “Window” призначена для: налаштування активних вікон, що відображаються в IDE (такі як: вікно з ієрархією файлів проекту, за замовчуванням відображається розгорнуте в лівій частині IDE, вікно, в якому відображається структура відкритого файлу з програмним кодом, що включає в себе: класи, методи, функції, зазвичай відображається в лівому нижньому куті IDE, вікно з конфігурацією файлів баз даних о підключені до проекту, зазвичай відображається у правій частині IDE), керування сповіщеннями в IDE, керування фоновими процесами, що виконуються в IDE та переключення між декількома одночасно запущеними проектами.

Вкладка “Help” - призначена для відкриття веб-сторінки з актуальною документацією, на якій програміст може знайти статті, в яких він може знайти відповіді на ті чи інші необхідні йому питання. Також в даній вкладці є опція, що дозволяє програмістові створити онлайн запит на лінію підтримки. Додатково присутні такі опції запуску діагностичних інструментів, перегляд логів за допомогою файлового провідника та інші, рис. 2.12.

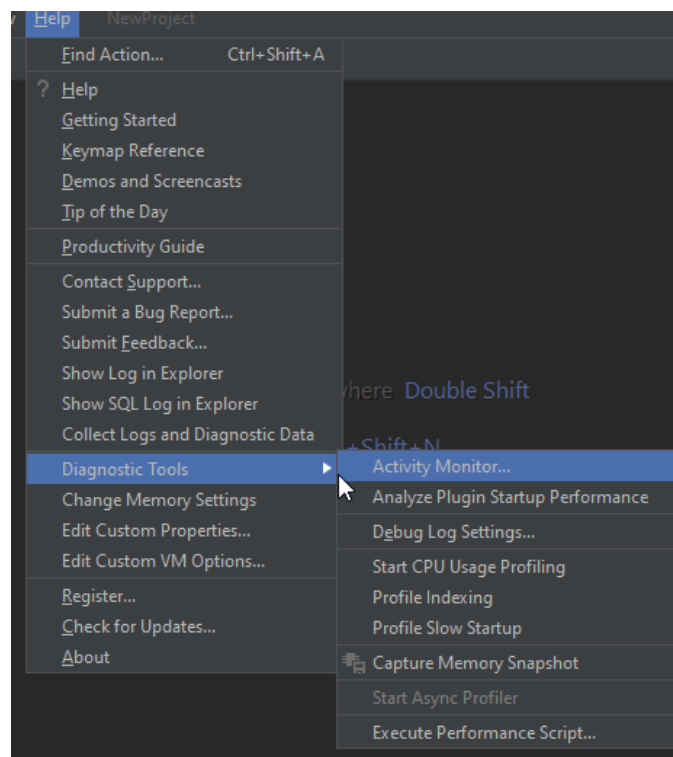


Рисунок 2.12. - перелік доступних опцій у вкладці “Help”.

2.4. Аналіз процесу створення чат-боту в месенджері «Telegram».

Для створення власного бота в месенджері Telegram, необхідно спочатку його зареєструвати. Реєстрація бота полягає в послідовному виконанні наступних операцій:

1. Створення нового бота за допомогою BotFather.
2. Отримання токенів доступу до створеного бота.
3. Редагування профілю бота за допомогою BotFather.
4. Авторизація та подальше програмування бота за допомогою отриманого токена.[9]

Створення нового бота полягає, в операції реєстрації бота на веб-сервері месенджера Telegram. Для того щоб його зареєструвати необхідно скористатись головним ботом месенджера - BotFather.

BotFather - бот, створений командою месенджеру Telegram, мета якого полягає у створенні нових, редагуванні та видаленні уже існуючих ботів, за допомогою вбудованих команд.

Таблиця 2.1. - Перелік загальнодоступних команд BotFather:

Команда	Опис команди.
/newbot	Створення нового бота.
/mybots	Керування створеними ботами.
/setname	Зміна імені бота.
/setdescription	Зміна опису бота.
/setabouttext	Зміна інформації про бота.
/setuserpic	Зміна зображення бота.
/setcommands	Зміна списку доступних команд.
/deletebot	Видалення бота.

Продовження таблиці 2.1

/token	Генерація токена доступу.
/revoke	Деактивація токена доступу.
/setinline	Керування підказками у рядку введення тексту.
/setinlinegeo	Керування запитами на розташування
/setinlinefeedback	Керування налаштуваннями зворотнього зв'язку
/setjoingroups	Чи можна бота додавати в групі чати?
/setprivacy	Керування режимом приватності в групових чатах.

Для ініціалізації діалогу з BotFather необхідно ввести ключову команду /start, рис. 2.13. Після ініціалізації діалогу бот надішле перелік доступних команд для керування чат-ботами, таблиця 2.1.

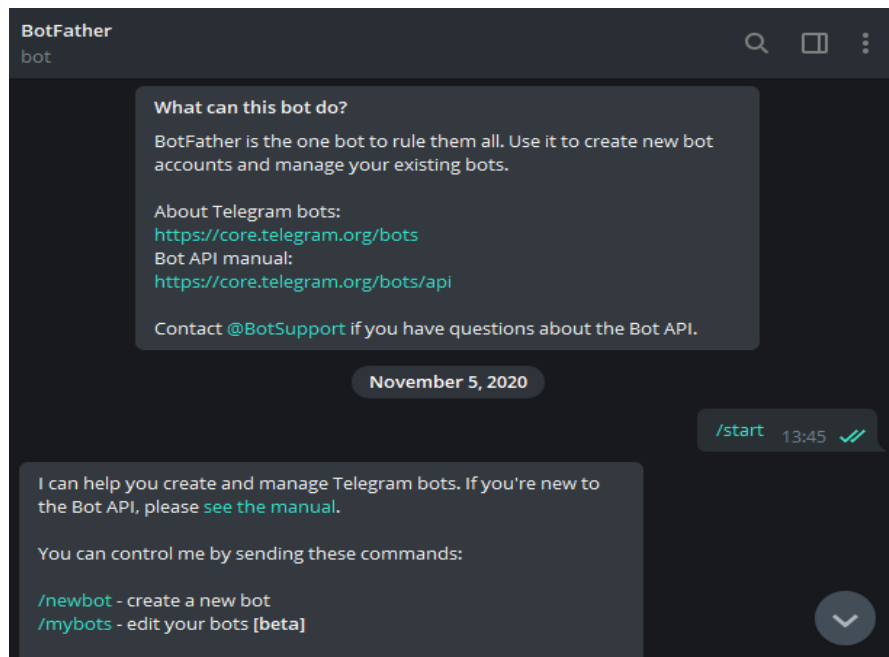


Рисунок 2.13. - ініціалізація діалогу з BotFather.

Тепер, скориставшись отриманим посиланням на чат-бота, є можливість перейти в його профіль, рис. 2.14, та ініціалізувати з ним діалог за допомогою команди /start.

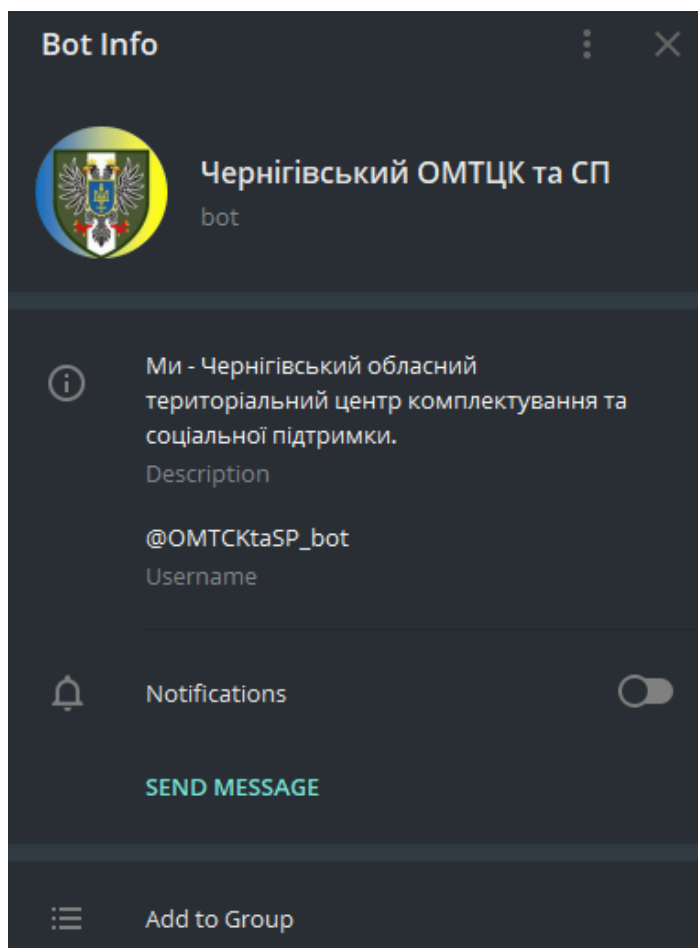


Рисунок 2.14. - профіль створеного чат-бота.

Подальше програмування чат-бота, залежить лише від ідеї та мети його створення, інакше кажучи, надалі програмувати чат-бота можна налюбій мові програмування, керуватись вибором якої необхідно в першу чергу уже набутиим досвідом програмування на ній, та в залежності від її функціональних можливостей, а саме чи може ця мова забезпечити реалізацію ідею та мети чат-бота.

Налаштування відправки запитів відбуваються за допомогою HTTPS запиту за адресою “<https://api.telegram.org/bot{token}/{methods}>”, де:

- {token} - токен авторизації чат-боту;
- {methods} - метод програмного коду, який формує та відправляє повідомлення, за умов що визначені у методі.

Для відправки запиту, доступні GET та POST запити. Основні властивості запитів наведені у таблиці 2.2.

Таблиця 2.2. - Основні властивості GET та POST запитів.

Властивість:	GET	POST
Призначення:	Запит на отримання певних даних.	Відправка певних даних, на обробку зі сторони серверу.
Кешування:	Кешуються браузером.	Ніколи не кешуються.
Історія перегляду:	Залишається відповідний запис.	Не збігаються.
Доступні кодування:	1. application/x-www-form-urlencoded	1. application/x-www-form-urlencoded. 2. multipart/form-data

2.5. Використання системи контролю версій Git та Github.

Git - безкоштовна система контролю версій програмного забезпечення, з відкритим вихідним кодом. Система призначена для зручного, швидкого та ефективного керування проектами різного обсягу[10].

Для того, щоб розпочати роботу з git, необхідно з офіційного сайту завантажити файл інсталятор, та встановити git на свій робочий ПК. Після встановлення, необхідно запустити git консоль, що має назву Git CMD.

Після першого запуску git на комп'ютері, необхідно налаштувати конфігураційні файли, а саме:

1. Вказати ім'я користувача.
2. Вказати електронну адресу користувача.

- Згенерувати ssh ключ доступу до репозиторію (у випадку коли репозиторій уже створено).

Створення нового репозиторію, відбувається за допомогою виконання команди `git init` у папці проекту, рис. 2.15.

```
Initialized empty Git repository in /home/user/Desktop/git_exercise/.git/
```

Рисунок 2.15. - результат виконання
команди `git init`.

Інформацію про стан репозиторію в даний момент часу можна отримати за допомогою команди `git status`, рис. 2.16.

```
$ git status
On branch master
Initial commit
Untracked files:
(use "git add ..." to include in what will be committed)
hello.txt
```

Рисунок 2.16. - результат виконання
команди `git status`.

Для того, щоб підготувати файли проекту до відправки в репозиторій, необхідно скористатись командою `git add -A`, де `-A` вказує, що необхідно підготувати всі файли, що знаходяться в даному репозиторії. Перевірити результат виконання команди `git add -A`, можна за допомогою команди `git status`. Для фіксації змін в певний момент часу - створюється коміти, за допомогою команди `git commit -m {"Commit description"}`.

Для підключення до віддаленого репозиторію, використовується команда `git remote add origin {URL}`, де `origin` - назва головного репозиторію, а `{URL}` - url адреса репозиторію.

Відправка змін на віддалений репозиторій виконується командою `git push origin master`, рис 2.17, `origin` - ім'я віддаленого репозиторію, `master` - гілка репозиторію, в яку необхідно внести зміни.

```
$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 212 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/tutorialzine/awesome-project.git
* [new branch] master -> master
```

Рисунок 2.17. - результат виконання команди `git push origin master`.

При одночасній роботі декількох розробників в одному проекті, необхідно час від часу синхронізувати локальний репозиторій із віддаленим, інакше кажучи, необхідно завантажити зміни із віддаленого репозиторію на локальний. Завантаження змін виконується за допомогою команди `git pull origin master`, рис.2.18.

```
$ git pull origin master
From https://github.com/tutorialzine/awesome-project
* branch master -> FETCH_HEAD
Already up-to-date.
```

Рисунок 2.18. - результат виконання команди `git pull origin status`.

В Git доступна можливість одночасно працювати з декількома гілками в одному репозиторії. Переваги роботи в декількох гілках:

1. Стабільна версія коду завжди зберігається у гілці `master`.
2. Різний функціонал одночасно може розроблятися декількома програмістами.

3. Програмісти можуть працювати у власній гілці, без страху що раптово хтось змінє програмний код в тому ж файлі.

Створення нової гілки відбувається за допомогою команди `git branch {name}`, де `{name}` - ім'я гілки. В результаті буде створено точну копію гілки `master`.

Переключення між гілками відбувається за допомогою команди `git checkout {branchname}`, де `{branchname}` - ім'я гілки на яку потрібно переключитися.

Для поєднання двох гілок одного проекту призначена команда `git merge {branchname}`, де `{branchname}` - ім'я гілки з якою потрібно поєднатися.

Інші часто використовувані команди наведені у таблиці 2.3.

Таблиця 2.3. - перелік часто використовуваних команд при роботі з Git[10]:

Назва команди	Опис
<code>git diff</code>	Використовується для знаходження різниці між двома Git гілками.
<code>git reset</code>	Використовується для відміни операції зміни файлу.
<code>git rm</code>	Використовується для видалення файлу з індекса робочої копії.
<code>git clean</code>	Команда для видалення сміття з каталогу.
<code>git log</code>	Команда для перегляду історії виконаних комітів.
<code>git stash</code>	Використовується для тимчасового збереження всіх незафіксованих змін.
<code>git fetch</code>	Використовується для локального збереження всіх змін з віддаленого сервера.
<code>git archive</code>	Використовується для створення архіву репозиторія, або вказаних файлів.
<code>git grep</code>	Використовується для пошуку заданого коду, або регулярного виразу по всьому репозиторію

3 ПРОЕКТУВАННЯ МАКЕТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Огляд програмних засобів для створення макетів

Макет - графічна схема, призначена для відображення всіх форм, полів, відступів, відтінків що будуть знаходитись на сторінці у схематичному відображенні, рис. 3.1.[11]

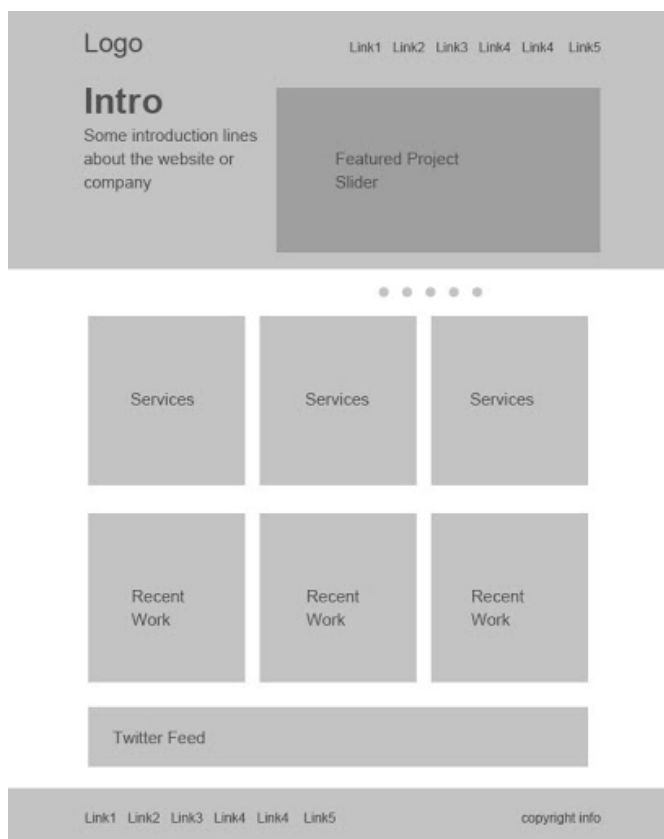


Рисунок 3.1. - приклад макету сайту.

Переваги використання макетів:

- Виявлення візуальних недоліків на ранніх стадіях роботи над проектом;
- Можливість наочно показати клієнту ідеї дизайнера;
- Спрощення роботи інших членів команди (верстальника та програміста).

Вимоги до готових макетів:

1. Симетричне відображення візуальних елементів.

2. Чітке визначення параметрів макета.
3. Створення елементів в різних комбінаціях відображення.
4. Використання окремих шарів для груп елементів.

Програмні засоби для створення макетів:

1. Wireframe.
2. Moqups.
3. Figma.
4. Adobe photoshop.
5. Adobe Illustrator.

Wireframe - онлайн середовище для створення макетів та прототипів сайтів з мінімальною кількістю вбудованих інструментів, рис 3.2.



Рисунок 3.2. - Робоча область Wireframe.

Особливості:

1. Мінімальний набір вбудованих інструментів, що максимально спрощує роботу з середовищем.
2. Можливість вибору уже з існуючих шаблонів розмірів робочої області (альбомні та портретні режими).
3. Можливість додавання коментарів з прив'язкою до об'єктів.

4. Смарт елементи, які за розміром вікна визначають тип елемента (заголовок, форма, скроллбар).

Moqups - зручний онлайн редактор для створення макетів та прототипів сайтів, рис. 3.3.

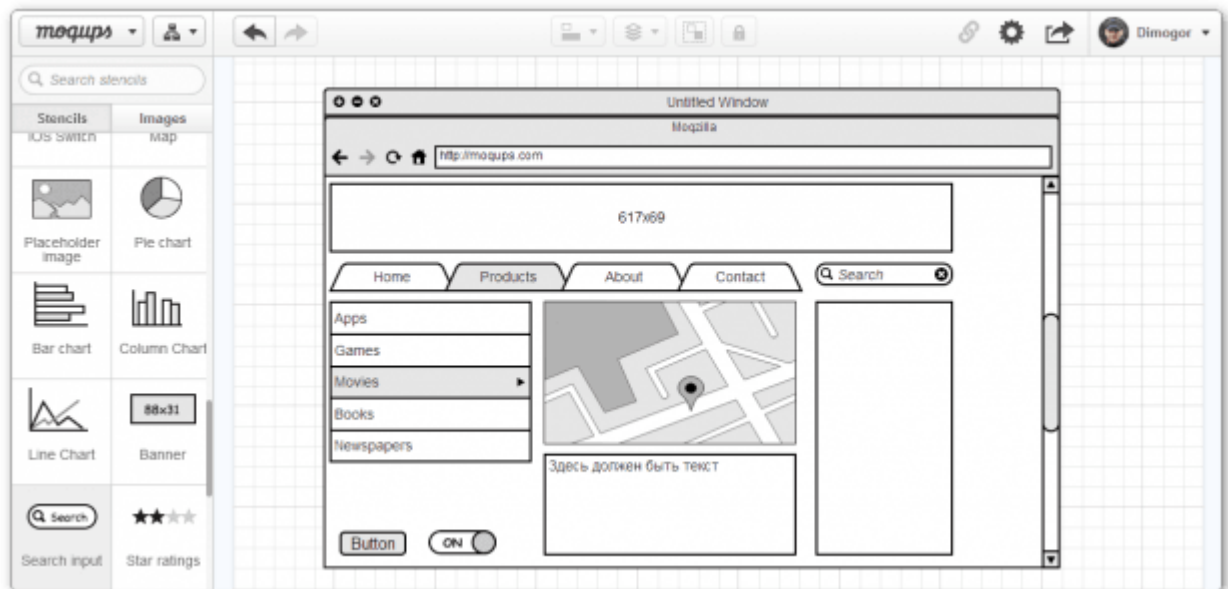


Рисунок 3.3. - Робоча область редактора Moqups.

Особливості:

1. Широкий вибір елементів для позначення кнопок, текстових блоків, заголовків, випадаючих списків та інших елементів.
2. Можливість пов'язати дві та більше сторінок з макетами.
3. Необмежена кількість редакторів, що працюють над макетами.
4. Підтримка роботи із зображеннями.
5. Експорт готових макетів у форматі PDF та PNG.

Figma - графічний редактор для веб-дизайну, призначений для створення інтерактивних прототипів сайтів та мобільних аплікацій, рис. 3.4. [12]

Особливості:

1. Збереження всіх макетів в хмарі.
2. Одночасне редагування макетів декількома редакторами.

3. Вбудована система контролю версій.
4. Підтримка роботи із зображеннями.
5. Можливість додавання коментарів з прив'язкою до об'єктів.
6. Режим презентації, в якому є можливість програти макет.
7. Вбудована бібліотека компонентів.
8. Необмежена кількість сіток, для зручної роботи з макетами.

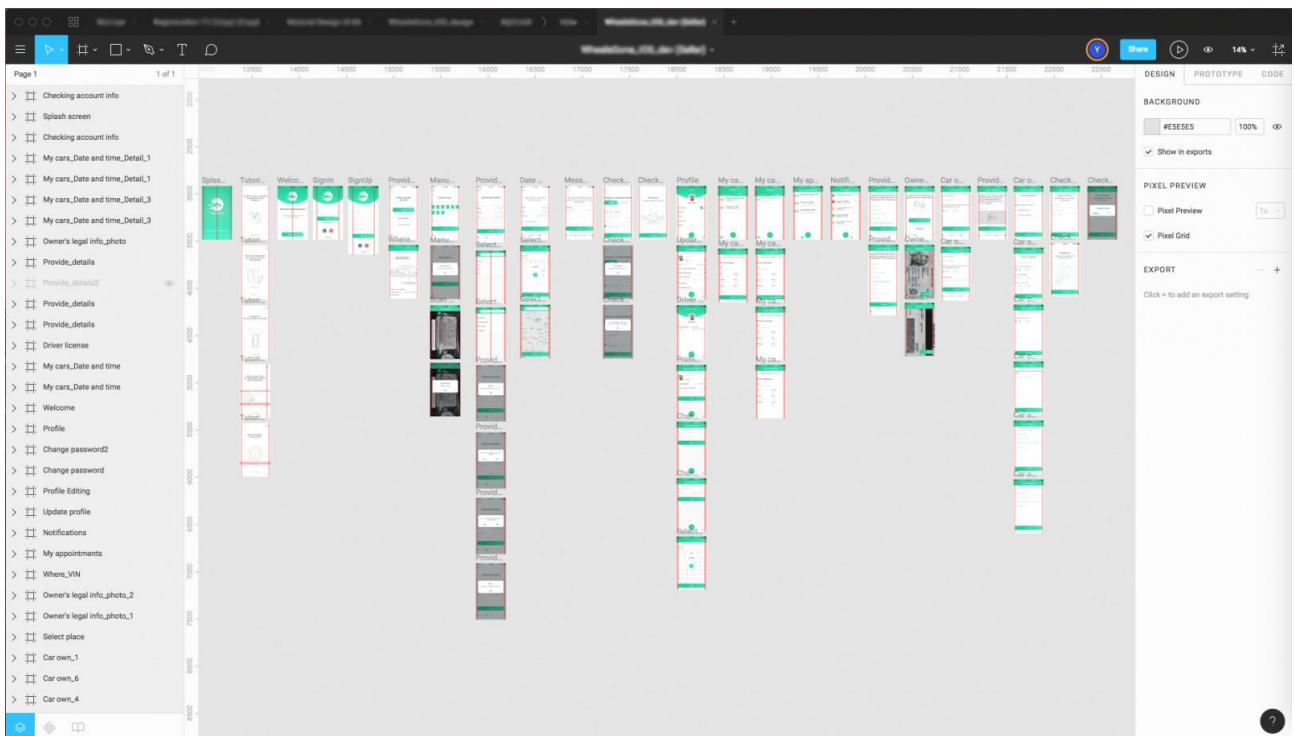


Рисунок 3.4. - Робоча область Figma.

Adobe Photoshop - багатофункціональний редактор призначений для роботи з фото та відео матеріалами, рис 3.5.[13]

Особливості:

1. Покращення якості зображень та відсканованих документів.
2. Створення багатошарового зображення, з можливістю окремого редагування кожного шару.
3. Відновлення старих зображень.
4. Можливість працювати з ескізами та кресленнями.
5. Широкий асортимент вбудованих інструментів для роботи з зображеннями.

6. Можливість експортувати макет у форматі PDF, PNG, JPG, JPEG та інших.

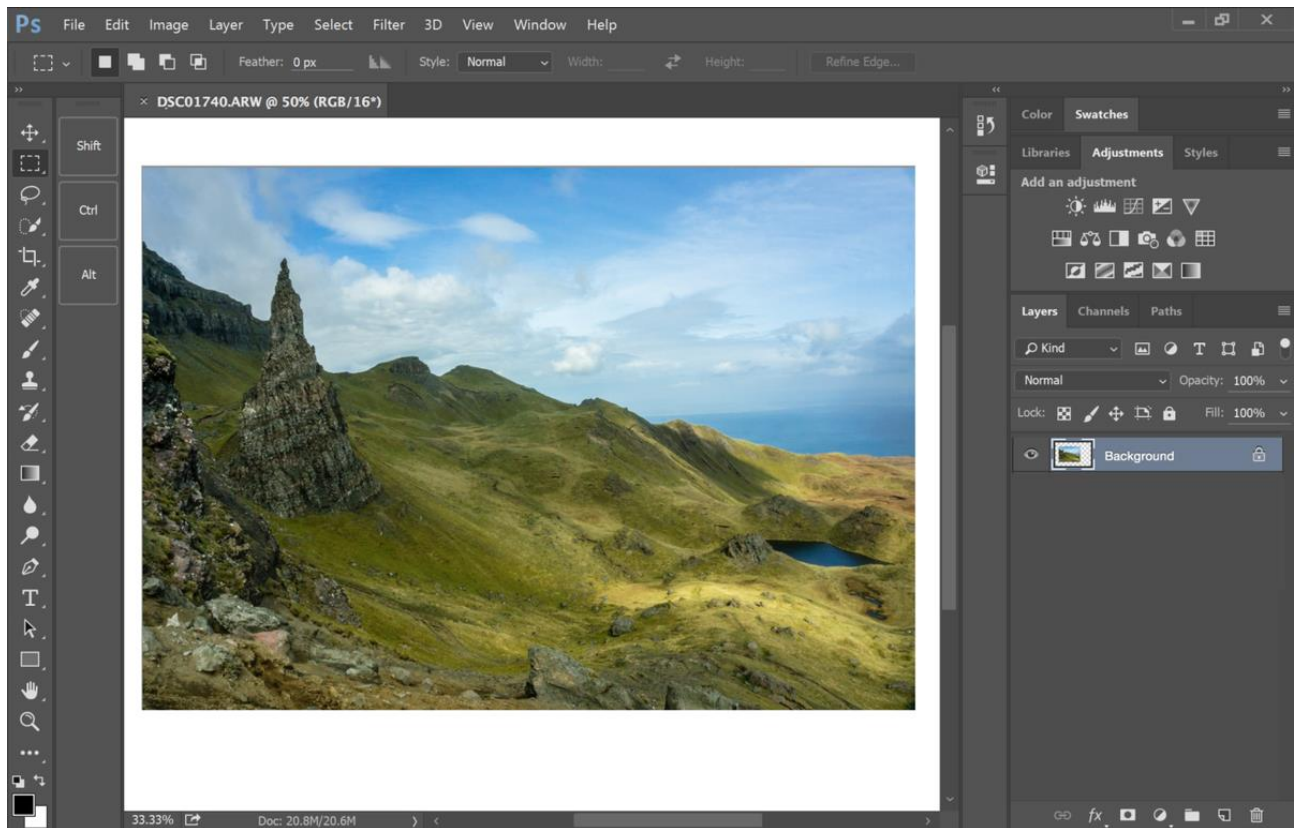


Рисунок 3.5. - Інтерфейс програми Adobe Photoshop.

Adobe Illustrator - програма, що використовується для створення векторної та веб графіки, рис. 3.6. На відміну від Adobe Photoshop, який працює з растровою графікою - Adobe Illustrator працює з векторною графікою[13].

Особливості:

1. Широкий набір інструментів для створення векторної графіки.
2. Експорт макетів у форматі PDF, PNG, JPEG, SVG та інших.
3. Створення багат шарового макету, з можливістю редагування кожного шару.
4. Побудова ліній за допомогою математичних формул.
5. Перетин та об'єднання декількох фігур.
6. Інструменти для створення логотипів та іконок.
7. Можливість використання раніше створених шаблонів.



Рисунок 3.6 - інтерфейс програми Adobe Illustrator.

3.2. Створення макетів інформаційної системи

Проаналізувавши програмні засоби для створення макетів, було прийнято за основний засіб Adobe Illustrator завдяки його особливостям перерахованим в розділі 3.1.

Перед початком роботи необхідно завантажити з офіційного сайту безкоштовну пробну версію та інсталиувати її.

Після запуску програми, користувачу на вибір пропонується:

1. Створити новий файл у шаблонних розмірах, або інших розмірах (в такому разі користувачу доведеться вручну задати розміри полотна).
2. Продовжити роботу над нещодавно закритими файлами.

Після створення нового файлу, завантажуються робоча область рис.3.7, в якій за замовчуванням відображається:

1. Панель інструментів.
2. Область для роботи із макетом.
3. Властивості макету.

4. Вікно для роботи із шарами.
5. Додаткові панелі можна відобразити за допомогою вкладки меню “Вікно”.

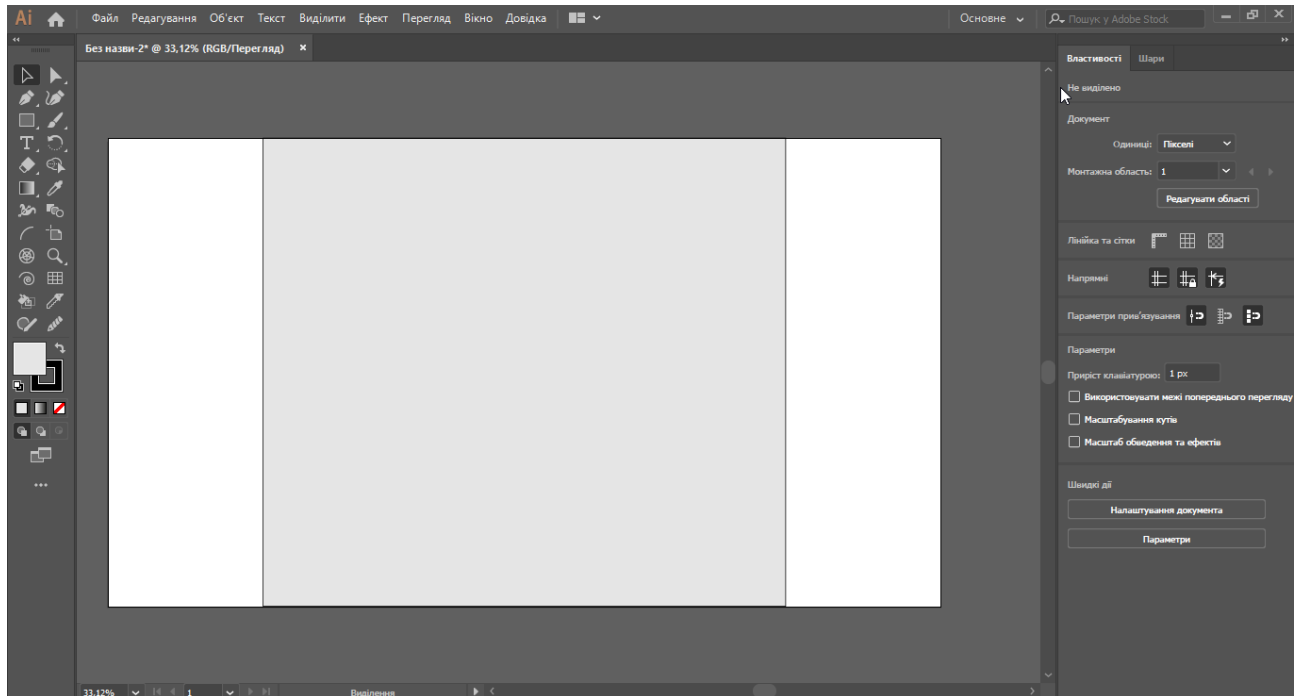


Рисунок 3.7. - Робоча область Adobe Photoshop.

Подальша робота зі створення макету відбувається наступним чином:

1. Створення документу, в якому будуть записані всі поля що повинні бути присутні на сторінці.
2. Розписується функціональне призначення полів[19].
3. За допомогою вбудованих інструментів Adobe Illustrator створюється макет.

Опис полів, що повинні бути присутні на сторінці “Шаблони повідомлень”:

1. Сайдбар, в якому перераховано перелік розділів в інформаційній системі.
2. Текстове поле із назвою активного розділу.
3. Кнопка для створення нового шаблону, після натиснення на неї - відкривається сторінка створення шаблону повідомлення.
4. Форма, призначена для пошуку шаблону, складається з:
 - a. Підпису по якому критерію проводити пошук.
 - b. Поля для вводу ключового слова, по якому здійснювати пошук.

- с. Кнопка фільтрації, після натиснення на яку, відбувається фільтрація шаблонів.
5. Таблиця, з відображенням уже існуючих шаблонів, містить наступні поля:
- а. Порядковий номер шаблону за списком.
 - б. ID шаблону в базі даних.
 - с. Назва шаблону, при натисненні відбувається перехід на сторінку редагування шаблону.
 - д. Період дії шаблону у форматі мм/чч/гг/хв, де:
 - і. мм - місяць.
 - ii. чч - число місяця.
 - iii. гг - година.
 - iv. хв - хвилина.
 - е. Кнопка видалення шаблону.

Завдяки детально опису всіх сторін полів та форм що повинні бути присутні на сторінці “Шаблони повідомлень”, було створено наступний макет, рис. 3.8.

The screenshot shows a web interface for managing message templates. On the left is a sidebar with five sections labeled 'Розділ 1' through 'Розділ 5'. The main content area is titled 'Шаблони повідомлень' and contains a 'Створити шаблон' button. Below this is a search area with a 'Назва' input field and a 'Фільтрувати' button. At the bottom is a table with the following data:

№	Код	Назва шаблону	Період	Дія
1	1	Шаблон 1	11/30/20/08	<input type="button" value="Видалити"/>
2	2	Шаблон 2	11/30/20/08	<input type="button" value="Видалити"/>

Рисунок 3.8. - Макет сторінки “Шаблони повідомлень”.

4 ІНСТРУКЦІ ЩОДО ВИКОРИСТАННЯ СИСТЕМИ

4.1. Інструкція менеджера системи

Адміністративна панель інформаційної системи, містить наступні розділи:

1. “Доступи”.
2. “Статична інформація”.
3. “Список призовників”.
4. “Шаблони повідомлень”.
5. “Повідомлення”.
6. “Історія повідомлень”.

Зміст та призначення кожного з розділів:

Розділ “Доступи” - призначений для створення, редагування та видалення доступів до адміністративної панелі інформаційної системи, рис. 4.1.

Доступи

Створити запис

Фільтр
Email
Фільтрувати

#	Код	Email	Активність	Дія
1	26	chernigivskiy_omtck_ta_sp@ukr.net	Активний	Видалити
2	25	goncharuk11@ukr.net	Активний	Видалити
3	24	onegodenko@gmail.com	Активний	Видалити
4	1	superadmin@admin.com	Активний	Видалити
5	20	vbondar@gmail.com	Активний	Видалити
6	23	vdziuba@gmail.com	Активний	Видалити
7	13	ziozyun@gmail.com	Активний	Видалити

Рисунок. 4.1. - Зображення розділу “Доступи”.

За замовчуванням в даному розділі відображається перелік уже наданих доступів у вигляді таблиці, що містить:

1. Порядковий номер відображення.

2. ID доступу в базі даних.
3. Електронна адреса, на яку наданий доступ.
4. Активність доступу (активний/неактивний).

Також в розділі “Доступи” є можливість виконати наступні дії:

1. Створення нового доступу.
2. Редагування існуючого доступу.
3. Видалення існуючого доступу.
4. Пошук доступу за його електронною адресою.

Сторінка створення та редагування доступу, рис. 4.2., містить наступні, необхідні для заповнення поля:

1. Електронна адреса для надання доступу.
2. Пароль для надання доступу.
3. Статус активності (активний/неактивний доступ).

Рисунок 4.2. - Сторінка редагування доступу

Розділ “Статична інформація” - призначений для керування статичною інформацією, що буде відображатись в профілі чат боту в месенджері “Telegram”, рис. 4.3.

При переході в даний розділ, менеджеру відображаються наступні поля:

1. Контактна інформація - текст з даного поля, буде надісланий користувачеві в месенджер “Telegram”, відразу після натискання користувачем кнопки “Контактна інформація”.

2. Повідомлення про успішну підписку - текст з даного поля, буде надісланий користувачеві в месенджер “Telegram”, відразу після успішної підписки на чат бота.
3. Кнопка “Редагувати” - при натисканні на дану кнопку, відбудеться збереження внесених змін для розділу “Статична інформація”.

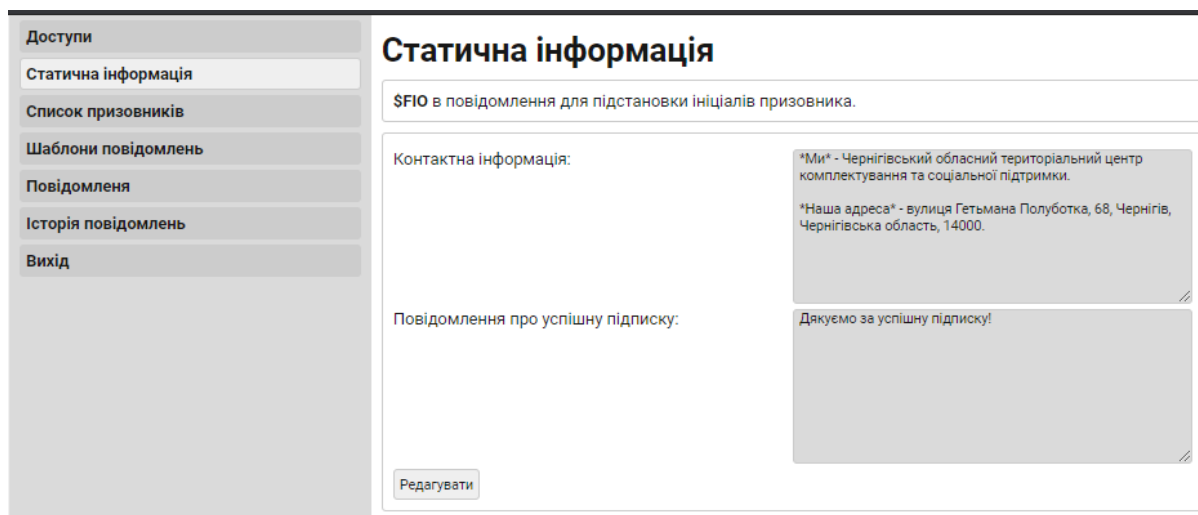


Рисунок 4.3. - Розділ “Статична інформація”

Розділ “Список призовників” - призначений для додавання, редагування або видалення інформації про призовника, рис. 4.4. Даний розділ необхідний для ідентифікації призовника за наступною схемою взаємодії:

1. Менеджер адмін. панелі інформаційної системи вносить інформацію про призовника.
2. Призовник в месенджері “Telegram”, знаходить чат бота, натискає кнопку “Start”.
3. Чат бот надсилає призовнику повідомлення із проханням ввести його ПІП.
4. Призовник вводить ПІП та відправляє його чат боту.
5. Після отримання повідомлення з ПІП можливі наступні варіанти розвитку подій:

- a. Якщо ППП співпало з тим що є в адмін. панелі - чат бот надсилає призовнику повідомлення про успішну підписку з розділу “Статична інформація”
- b. Якщо ППП не співпало з тими що є в адмін. панелі - чат-бот просить повторити повідомлення з ППП призовника.

Доступи

Статична інформація

Список призовників

Шаблони повідомлень

Повідомлення

Історія повідомлень

Вихід

Список призовників

Створити запис

Фільтр

Ініціали

Номер телефону

Фільтрувати

#	Код	Ініціали	Номер телефону	Дата народження	Дія
1	9	Бондар Василь Вікторович	0678358337	1999-06-16 00:00:00	Видалити
2	15	Володько Вадим Геннадійович	0637829848	1980-02-23 00:00:00	Видалити
3	12	Гончарук Олег Валентинович	+380955472803	1975-03-11 00:00:00	Видалити
4	10	Дзюба Вадим Віталійович	+380000000000	2020-12-27 00:00:00	Видалити
5	8	Ліщук Ігор Валерійович	+0000000000	1999-08-22 00:00:00	Видалити
6	16	Луговський Олександр Васильович	0504657343	1973-04-09 00:00:00	Видалити
7	17	Луговський Олександр Васильович	0504657343	1973-04-09 00:00:00	Видалити
8	11	Негоденко Олена Василівна	+380000000000	2020-12-27 00:00:00	Видалити
9	14	Парамонова Олена Вікторівна	0953093659	1980-02-23 00:00:00	Видалити
10	13	Прізвище ім'я По батькові	+38000000	2021-01-12 00:00:00	Видалити

Рисунок 4.4. - Розділ “Список призовників”

За замовчуванням в даному розділі відображається перелік уже доданої короткої інформації про призовників у вигляді таблиці, що містить:

1. Порядковий номер відображення.
2. ID доступу в базі даних.
3. Прізвище Ім'я По-батькові призовника.
4. Номер телефону призовника.
5. Дата народження призовника.

Також в розділі “Список призовників” є можливість виконати наступні дії:

1. Додати інформацію про нового призовника.
2. Редагувати інформацію про існуючого призовника.
3. Видалення інформації про існуючого призовника.
4. Пошук інформації про призовника за його електронною адресою.
5. Пошук інформації про призовника за його ПІБ.

Сторінка створення та редагування інформації про призовника, рис. 4.5., містить наступні, необхідні для заповнення поля:

1. Прізвище.
2. Ім'я.
3. По-батькові.
4. Номер телефону.
5. Дата народження.
6. Кнопка “Редагувати” - при натисканні на дану кнопку, відбудеться збереження внесених змін в інформацію про призовника.

Рисунок 4.5. - Редагування інформації про призовника.

Розділ “Шаблони повідомлень” - призначений для створення, редагування та видалення шаблонів повідомлень, які будуть надіслані в запланований час та лише вибраним в шаблоні призовникам, в чат-боті месенджера “Telegram”, рис. 4.6.

За замовчуванням в даному розділі відображається перелік уже створених шаблонів в у вигляді таблиці, що містить:

1. Порядковий номер відображення.
2. ID шаблону в базі даних.

3. Назва шаблону.
4. Період надсилання шаблону.

Список шаблонів

Створити запис

Фільтр
Назва

Фільтрувати

#	Код	Назва	Період	Дія
1	4	Тестовий	12/25/19/22	Видалити

Рисунок 4.6. - Розділ “Шаблони повідомлень”.

Також в розділі “Шаблони повідомлень” є можливість виконати наступні дії:

1. Створення нового шаблону повідомлення.
2. Редагування існуючого шаблону повідомлення.
3. Видалення існуючого шаблону повідомлення.
4. Пошук шаблону повідомлення за його назвою.

Сторінка створення та редагування шаблону повідомлення, рис. 4.7.

Редагування шаблону повідомлення

* - будь-яке число / місяць.
SFIO в повідомлення для підстановки ініціалів призовника.

Активність:

Назва:

Повідомлення:

Місяць:

День:

Година:

Хвилина:

Призовники:

Всі
 Бондар Василь Вікторович
 Ліщук Ігор Валерійович
 Луговський Олександр Васильович
 Негоденко Олена Василівна
 Парамонова Олена Вікторівна

Редагувати

Рисунок 4.7. - Створення шаблону повідомлення

Список присутніх полів на сторінці створення шаблону повідомлень:

1. Активність шаблону.
2. Назва шаблону.
3. Повідомлення, яке необхідно надіслати.
4. Вибір часу та дати надсилання повідомлення, складається з полів:
 - a. Місяць.
 - b. День.
 - c. Година.
 - d. Хвилина.
5. Вибір призовників, яким буде надіслано повідомлення, має можливі варіанти:
 - a. “All” - всім призовникам без винятку.
 - b. Вибір декількох призовників із переліку.
6. Кнопка “Додати” - після натискання, відбувається валідація всіх обов’язкових полів, і в разі успішної валідації - відбувається збереження всіх внесених в шаблон змін.

Розділ “Повідомлення” - призначений для створення та надсилання одноразового повідомлення, вибраним в налаштуваннях призовникам, в чат-боті месенджера “Telegram”, рис. 4.8.

Надсилання повідомлень

\$FIO в повідомлення для підстановки ініціалів призовника.

Повідомлення:

Призовники:

- Всі
- Бондар Василь Вікторович
- Ліщук Ігор Валерійович
- Луговський Олександр Васильович
- Негоденко Олена Василівна
- Парамонова Олена Вікторівна

Надіслати

Рисунок 4.8. - Сторінка створення повідомлення

За замовчуванням в даному розділі відображається сторінка створення повідомлення, яка містить наступні елементи:

1. Поле для введення тексту повідомлення.
2. Вибір призовників, яким буде надіслано повідомлення, має можливі варіанти:
 - a. “All” - всім призовникам без винятку.
 - b. Вибір декількох призовників із переліку.
3. Кнопка “Надіслати” - при натисканні буде надіслано повідомлення вибраним призовникам.

Розділ “Історія повідомлень” - призначений для відображення наступної інформації по кожному надісланому повідомленню, рис.4.9.:

1. Порядковий номер відображення.
2. Номер повідомлення в базі даних.
3. ID користувача, що надіслав повідомлення.
4. ID отримувача - призовник, якому було надіслано повідомлення.
5. Повідомлення - текст, який було надіслано призовникам.
6. Дата відправлення повідомлення, у форматі мм/чч/гг/хв, де:
7. Кнопки переходу на наступну/попередню сторінки.

Історія повідомлень						
#	Код	ID користувача	Отримувачі	Повідомлення	Дата відправлення	
1	27	25	['14']	ок	2/26/8/36	
2	26	25	['14']	ок	2/26/8/35	
3	25	25	['14']	шо	2/26/8/35	
4	24	25	['all']	як справи	2/26/8/33	
5	23	25	['14']	Привіт	2/26/8/32	
6	22	25	['16']	привіт	2/26/8/21	
7	21	25	['16']	доброго ранку	2/26/8/20	
8	20	25	['16']	вітаю	2/23/12/55	
9	19		['all']	SFIO Тест історії	12/25/19/22	
10	17	13	['9']	Тест історії	12/20/13/58	
11	16	20	['all']	тест	12/20/13/26	
12	15	20	['4']	тест	12/20/13/25	

Рисунок 4.9. - Розділ “Історія повідомлень”

4.2. Інструкція кінцевого користувача

Для використання чат-бота в месенджері Telegram, необхідно:

1. Знайти чат-бот за іменем користувача “@OMTCKtaSP_bot” в месенджері Telegram.
2. У вікні повідомлень необхідно натиснути кнопку “Start”, після чого чат-бот запитає Прізвище Ім’я По-батькові призовника.
3. ПІП введений призовником, перевіряється із наявними даними в списку призовників. І в разі успішного проходження перевірки, користувач отримує ThankYouMessage про успішну підписку.

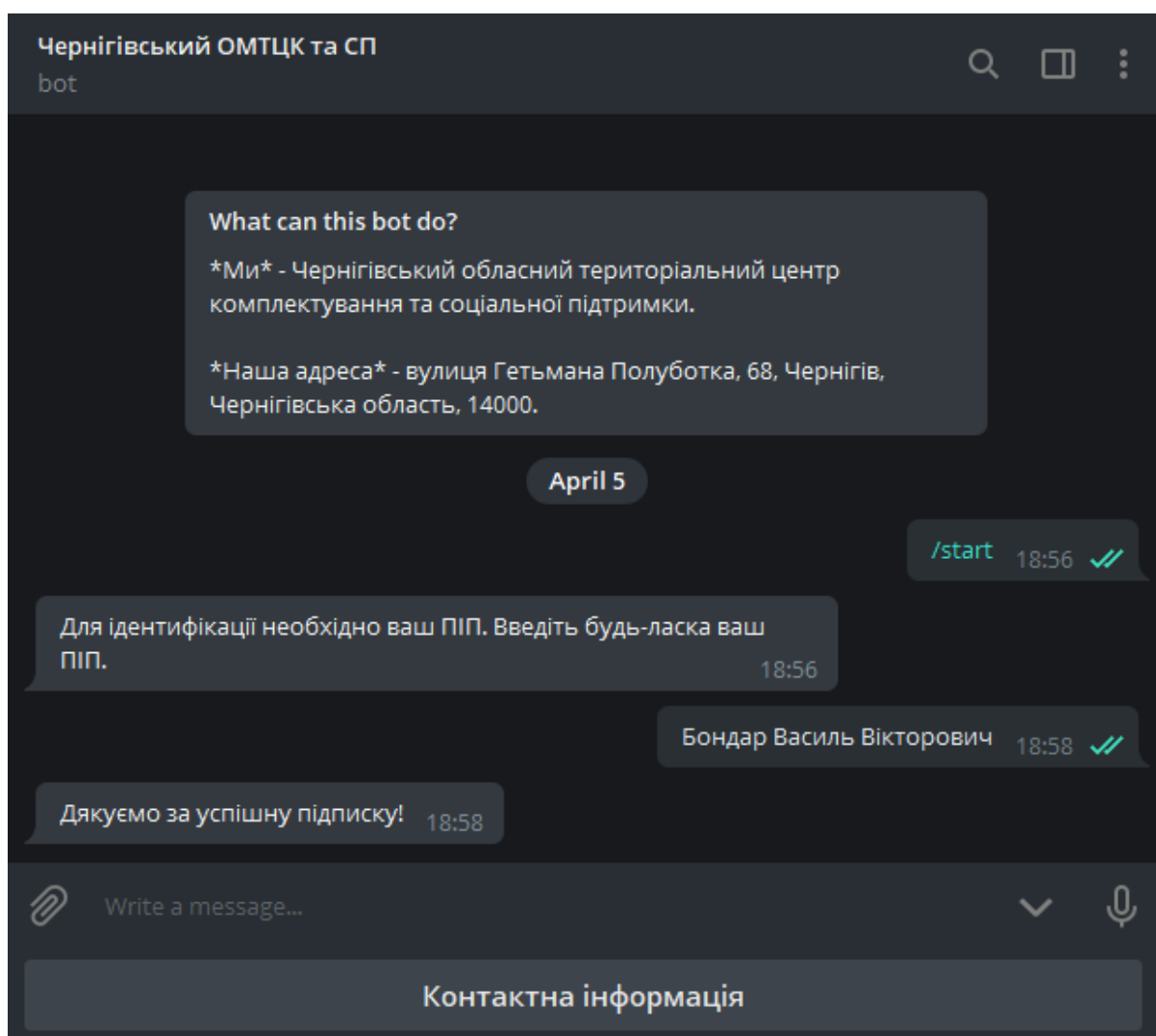


Рисунок 4.10. - Процес ідентифікації призовника.

Після процесу ідентифікації, рис. 4.10., призовник автоматично буде отримувати всі надіслані повідомлення з інформаційної системи.

4.3. Тестування інформаційної системи

Розроблена інформаційна система має працювати в режимі 24/7, тобто цілодобово та без збоїв. Під час тестування, було застосовано методи ручного та автоматизованого тестування, а також складено перелік тест кейсів для кожного з розділів системи.

Метод ручного тестування полягає в виконанні тестувальником тест-кейсів без використання програмних засобів, для перевірки веб сайту шляхом виконання дій користувача.

Автоматизоване тестування полягає в написанні скриптів згідно розроблених тест-кейсів, в результаті виконання яких формується звіт з переліком успішних та неуспішних тест кейсів.

Тест кейс - артефакт, в якому описано сукупність кроків та параметрів, необхідних для перевірки функціоналу, що тестується[14]. Приклад тест кейсу наведено у таблиці 4.1.

Під тест кейсом розуміється наступна ієрархія виконання дій: Дія -> Очікуваний результат -> Фактичний результат, де фактичний результат не завжди дорівнює очікуваному, та навпаки[15].

В свою черги тест кейси поділяються на:

- Позитивні;
- Негативні.

Позитивні тест кейси використовуються лише з коректним набором даних і призначені для перевірки коректності роботи функції, що виконується.

Негативні тест кейси використовуються як з коректним так завчасно некоректним набором даних, і має на меті перевірити роботу функції що виконується у випадку виняткової ситуації[16].

Таблиця 4.1 - Тест кейси для тестування функціоналу ідентифікації призовника

Функціонал ідентифікації призовника		
<p>Передумови:</p> <ol style="list-style-type: none"> 1. Призовник вже має аккаунт месенджера Telegram. 2. Призовник уже ініціалізував діалог з чат ботом. 		
Крок тесту	Дані тесту	Очікуваний результат
Ідентифікація призовника	<p>Створити відповідний запис у інформаційній системі, в розділі “Список призовників”.</p> <ul style="list-style-type: none"> - Надіслати ППІ призовника чат ботові. 	<ul style="list-style-type: none"> - Чат бот наділе повідомлення про успішну підписку.
Ідентифікація призовника з некоректним ППІ	<ul style="list-style-type: none"> - Створити відповідний запис у інформаційній системі, в розділі “Список призовників”. - Надіслати некоректний ППІ призовника чат ботові. 	<ul style="list-style-type: none"> - Чат бот надішле повідомлення що Призовника з таким ППІ не знайдено.

ВИСНОВКИ

Робота присвячена створенню інформаційної системи сповіщення для покращення комунікації співробітників військкомату та призовників. В процесі дослідження теми та мети було проведено:

1. Проведено дослідження, які обґрунтовують наукову новизну та актуальність роботи, а саме: встановлено, що за допомогою об'єктно орієнтованих мов програмування, та інтегрованих середовищ програмування було досягнуто поставленої мети.
2. За допомогою існуючих методик визначення вимог до ПЗ вдалось чітко сформулювати ключові вимоги до інформаційної системи, та встановити їх пріоритети.
3. Огляд та аналіз існуючих месенджерів на території України, вдалось їх класифікувати за рядом ознак, та встановити що месенджер «Telegram» повністю задовольняє визначені вимоги.
4. Завдяки чіткому описові популярних на сьогодні мов програмування, було визначено що мова програмування PHP в комбінації з IDE від компанії JetBrains повністю задовольняє встановлені вимоги до інформаційної системи.
5. На етапі створення макетів інформаційної системи, було виявлено деякі її недоліки. Проте завдяки застосуванню водоспадної моделі розробки програмного забезпечення, знадобилось мінімум часу для перегляду вимог з метою усунення виявлених недоліків.

На закінчення можна зробити висновок: інформаційна система, здатна значно спростити процес комунікації військкомату з призовниками, тому її рекомендується використовувати як додатковий засіб інформування призовників про початок призову та в інших ситуаціях.

Регулярне використання системи дозволить зробити процес комунікації між співробітниками військкомату та призовниками більш простим та прозорішим, на відміну від того, що є сьогодні.

СПИСОК ЛІТЕРАТУРИ

1. jauxt.github.io [Електронний ресурс] : [Інтернет-портал] - Електронні дані. [US, Міжнародна організація “GitHub, Inc.”]. - Режим доступу <https://jauxt.github.io/MessengerComparison> (дата звернення 10.04.2021). - Назва з екрана.
2. Грицюк Юрій, Аналіз вимог до програмного забезпечення. //Львівська політехніка. -2018. 456с.
3. Роберт Мартін, Чиста архітектура: мистецтво розробки програмного забезпечення. // Фабула. - 2019. 416с.
4. Роберт Сесіл Мартін, Чистий код. Створення і рефакторинг за допомогою Agile // Фабула. -2019. 448с.
5. Mike McGrath C# Programming in Easy Steps // In Easy Steps. - 2020, 192с.
6. Олексій Васильєв Програмування мовою Python // Навчальна книга - Богдан - 2019. 504с.
7. William Sanders, Learning PHP Design Patterns // O'Reilly Media. - 2013. 362с
8. www.jetbrains.com [Електронний ресурс] : [Інтернет-портал] - Електронні дані. [Jacksonville, Міжнародна організація “PERFECT PRIVACY, LLC”]. - Режим доступу <https://www.jetbrains.com/phpstorm/documentation/> (дата звернення 11.04.2021). - Назва з екрана.
9. <https://core.telegram.org/bots/api> [Електронний ресурс] : [Інтернет-портал] - Електронні дані. [Arizona, Приватна організація “Telegram, Inc”]. - Режим доступу <https://core.telegram.org/bots/api> (дата звернення 12.04.2021). - Назва з екрана.
10. Скотт Чакон та Бен Страуб, ProGit друге видання // Apress. - 2014. 529с
11. Jeff Gothelf Josh Seiden, Lean UX. Applying Lean Principles to Improve User Experience // O'Reilly Media. - 2013, 152с

12. <https://www.figma.com/> [Електронний ресурс] : [Інтернет-портал] - Електронні дані. [Reykjavik, Приватна організація “Figma Inc”]. - Режим доступу <https://www.figma.com/> (дата звернення 14.04.2021). - Назва з екрана.
13. <https://www.adobe.com/> [Електронний ресурс] : [Інтернет-портал] - Електронні дані. [San Jose, Публічна організація “Adobe Inc”]. - Режим доступу <https://www.adobe.com/> (дата звернення 15.04.2021). - Назва з екрана.
14. Корі Сандлер, Гленфорд Майерс, Том Баджетт, Мистецтво тестування програм // Діалектика – 2016, 272с.
15. William E. Lewis, Software Testing and Continuous Quality Improvement // CRC Press – 2017, 688с.
16. Rex Black, Advanced Software Testing - Vol. 1, 2nd Edition // Rocky Nook – 2015, 376с.
17. Sameer Paradkar, Mastering Non-Functional Requirements // Packt Publishing Ltd – 2017, 230с.
18. Phillip A. Laplante, Requirements Engineering for Software and Systems // CRC Press – 2017, 378с.
19. Akash Khandavilli, 100 Dos and Dont's UI/UX Design Tips eBook- Professional research based // Akash Khandavilli – 2020, 78с.

Додаток А

Програмний код файлу index.php

```
1. <?php
2. //ini_set('display_errors', 1);
3. //error_reporting(E_ALL);
4. //ini_set('display_startup_errors', 1);
5.
6. session_start();
7.
8. define('ROOT_PATH', __DIR__);
9.
10.    spl_autoload_register(function ($className) {
11.        $className = str_replace('\\', PATH_SEPARATOR, $className);
12.        $path = ROOT_PATH . "/helpers/{$className}.php";
13.        if (file_exists($path)) {
14.            require_once $path;
15.        }
16.    });
17.
18.
19.    if (isset($_GET['module'])) {
20.        $path = __DIR__ . "/modules/{$_GET['module']}.php";
21.        if (file_exists($path)) {
22.            require_once $path;
23.        }
24.    } else {
25.        $find = false;
26.        $url = $_GET['url'] == "" ? 'home' : $_GET['url'];
27.        if ($handle = opendir('controllers')) {
28.            while (false !== ($file = readdir($handle))) {
```



```
29.     $pathInfo = pathinfo($file);
30.     if ($pathInfo['extension'] == 'php') {
31.         require_once ROOT_PATH . "/controllers/{$file}";
32.         $className = $pathInfo['filename'];
33.         if (class_exists($className)) {
34.             $controller = new $className();
35.             /** @var $controller Controller */
36.             if ($controller->audit($url)) {
37.                 echo $controller->validate() ? $controller->start() :
                 $controller->error();
38.                 $find = true;
39.                 break;
40.             }
41.         }
42.     }
43. }
44. closedir($handle);
45. }
46. if (!$find && $url != 'home') {
47.     header('location: /');
48. }
49. }
```

Додаток Б

Програмний код сторінки створення шаблону повідомлення

```

1. <?php /** @var @data array */ $edit = isset($data['row']); ?>
2. <?php if (isset($data['success'])): ?>
3.     <div class="success"><?= $data['success'] ?></div>
4. <?php endif; ?>
5. <div class="filter">
6.     <b>*</b> - будь-яке число / місяць.<br>
7.     <b>$FIO</b> в повідомлення для підстановки ініціалів прізвища.
8. </div>
9. <form method="post" class="filter">
10.     <fieldset>
11.         <label for="active">Активність:</label>
12.         <select name="active" id="active">
13.             <option value="0" <?= ($data['row']['active'] ?? false) == false ?
'selected' : null ?>>Не активний</option>
14.             <option value="1" <?= ($data['row']['active'] ?? false) == true ?
'selected' : null ?>>Активний</option>
15.         </select>
16.     </fieldset>
17.     <fieldset>
18.         <label for="title">Назва:</label>
19.         <input id="title" name="title" value="<?= $data['row']['title'] ?? "
?>" type="text" placeholder="Назва" required>
20.     </fieldset>
21.     <fieldset>
22.         <label for="message">Повідомлення:</label>
23.         <textarea name="message" id="message" required><?=
$data['row']['message'] ?? " ?></textarea>
24.     </fieldset>

```

```

25.     <fieldset>
26.         <label for="month">Місяць:</label>
27.         <?php $month = $data['row']['mouth'] ?? ""; ?>
28.         <select name="month" id="month">
29.             <option value="*" <?= $month == '*' ? 'selected' : "
?>>*</option>
30.             <option value="1" <?= $month == '1' ? 'selected' : "
?>>Січень</option>
31.             <option value="2" <?= $month == '2' ? 'selected' : "
?>>Лютий</option>
32.             <option value="3" <?= $month == '3' ? 'selected' : "
?>>Березень</option>
33.             <option value="4" <?= $month == '4' ? 'selected' : "
?>>Квітень</option>
34.             <option value="5" <?= $month == '5' ? 'selected' : "
?>>Травень</option>
35.             <option value="6" <?= $month == '6' ? 'selected' : "
?>>Червень</option>
36.             <option value="7" <?= $month == '7' ? 'selected' : "
?>>Липень</option>
37.             <option value="8" <?= $month == '8' ? 'selected' : "
?>>Серпень</option>
38.             <option value="9" <?= $month == '9' ? 'selected' : "
?>>Вересень</option>
39.             <option value="10" <?= $month == '10' ? 'selected' : "
?>>Жовтень</option>
40.             <option value="11" <?= $month == '11' ? 'selected' : "
?>>Листопад</option>
41.             <option value="12" <?= $month == '12' ? 'selected' : "
?>>Грудень</option>
42.         </select>
43.     </fieldset>

```

```

44.     <fieldset>
45.         <label for="day">День:</label>
46.         <?php $day = $data['row']['day'] ?? ""; ?>
47.         <select name="day" id="day">
48.             <option value="" <?= $day == '*' ? 'selected' : "
?>>*</option>
49.             <?php for ($i = 1; $i <= 31; $i++): ?>
50.                 <option value=""<?= $i ?>" <?= $day == $i ? 'selected' : "
?>><?= $i ?></option>
51.             <?php endfor; ?>
52.         </select>
53.     </fieldset>
54.     <fieldset>
55.         <label for="hour">Година:</label>
56.         <?php $hour = $data['row']['hour'] ?? ""; ?>
57.         <select name="hour" id="hour">
58.             <?php for ($i = 0; $i <= 23; $i++): ?>
59.                 <option value=""<?= $i ?>" <?= $hour == $i ? 'selected' : "
?>><?= $i < 10 ? "0{$i}" : $i ?></option>
60.             <?php endfor; ?>
61.         </select>
62.     </fieldset>
63.     <fieldset>
64.         <label for="minute">Хвилина:</label>
65.         <?php $minute = $data['row']['minute'] ?? ""; ?>
66.         <select name="minute" id="minute">
67.             <?php for ($i = 0; $i <= 59; $i++): ?>
68.                 <option value=""<?= $i ?>" <?= $minute == $i ? 'selected' : "
?>><?= $i < 10 ? "0{$i}" : $i ?></option>
69.             <?php endfor; ?>
70.         </select>
71.     </fieldset>

```

```

72.     <fieldset>
73.         <span>Призовники:</span>
74.         <div class="filter">
75.             <label><input class="all" type="checkbox" name="conscripts[]"
value="all" <?= $edit ? count($data['row']['conscripts']) == 1 &&
$data['row']['conscripts'][0] == 'all' ? 'checked' : '' ?>>Всі</label>
76.             <?php foreach ($data['conscripts'] as $conscript): ?>
77.                 <label><input type="checkbox" name="conscripts[]"
value="<?= $conscript['id'] ?>" <?= $edit ? in_array($conscript['id'],
$data['row']['conscripts']) ? 'checked' : '' ?>><?= $conscript['initials']
?></label>
78.                 <?php endforeach; ?>
79.             </div>
80.         </fieldset>
81.         <button><?= $edit ? 'Редагувати' : 'Додати' ?></button>
82.     </form>

```

Додаток В



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
 ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ СПОВІЩАННЯ ДЛЯ ПОКРАЩЕННЯ РОБОТИ ВІЙСЬККОМАТУ З ВИКОРИСТАННЯМ ОБ'ЄКТНО-ОРІЄНТОВНИХ МОВ ПРОГРАМУВАННЯ»

Виконав студент 4 курсу
 Групи ПД-42
 Бондар Василь Вікторович

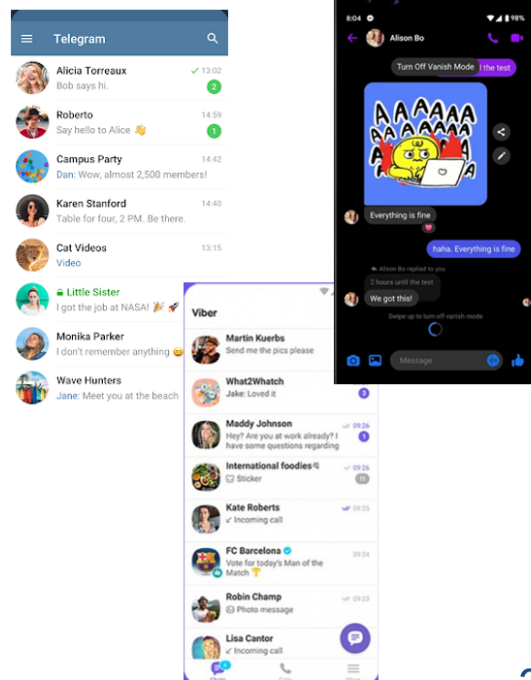
Київ – 2021

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- *Мета роботи* – підвищення ефективності сповіщення шляхом розробки інформаційної системи для покращення роботи військкомату з використанням об'єктно орієнтованих мов програмування.
- *Об'єкт дослідження* – процес передачі інформації для покращення роботи військкомату.
- *Предмет дослідження* – інформаційна система для покращення процесу надсилання сповіщень військкоматом.

АНАЛОГИ

	Telegram	Viber	WhatsApp
Компанія	Telegram FZ LLC	Rakuten, Inc.	Facebook
Тип	Некомерційний	Комерційний	Комерційний
Ідея	Свобода безпеки та приватності спілкування доступна для кожного	"Максимально захищений месенджер." Резервні копії не зашифровані.	"Конфіденційність та безпека – частина нашої ДНК." Резервні копії не зашифровані, виявлені вразливості.
Документація	Відкрита та деталізована	Відкрита та деталізована (розробка нових додатків неможлива)	Практично відсутня
Кількість Активних Користувачів	500 мільйонів	260 мільйонів	2000 мільйонів
Кількість Пристроїв	Необмежена	Один первинний та декілька вторинних пристроїв	Два



3

ТЕХНІЧНІ ЗАВДАННЯ

1. Надсилання повідомлень призовникам за заданим графіком.
2. Реалізація рольової моделі.
3. Логування усіх надісланих повідомлень.
4. Редагування статичної інформації за допомогою адмін. панелі.

4

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Telegram

5

Можливості адміністратора
інформаційної системи



Менеджер
системи



6

Візуальна частина роботи

The screenshot displays two main sections of a web application. On the left, a sidebar menu includes options like 'Доступи', 'Статична інформація', 'Список призовників', 'Шаблони повідомлень', 'Повідомлення', 'Історія повідомлень', and 'Вихід'. The main area is divided into two panels.

The top panel, titled 'Доступи', features a 'Створити запис' button, a search filter for 'Email', and a table with columns for '#', 'Код', 'Email', 'Активність', and 'Дія'. The table contains the following data:

#	Код	Email	Активність	Дія
1	26	chernigivsky_omtk-ta_sq@ukr.net	Активний	Видалити
2	25	goncharuk11@ukr.net	Активний	Видалити
3	24	onegodenko@gmail.com		
4	1	supradmin@admin.com		
5	20	ybondar@gmail.com		
6	23	yfzuba@gmail.com		
7	13	ziozyun@gmail.com		

The bottom panel, titled 'Редагування шаблону повідомлення', includes a note: '* - будь-яке число / місяць. \$FIO в повідомлення для підстановки ініціалів призовника.' It contains form fields for 'Активність:' (Active), 'Назва:' (Test), 'Повідомлення:' (SPIO), 'Місяць:' (Квітень), 'День:' (5), 'Година:' (18), and 'Хвилина:' (55). A 'Призовники:' field shows a list of names with checkboxes, including 'Бондар Василь Вікторович' (checked), 'Ліщук Ігор Валерійович' (checked), 'Луговський Олександр Васильович', 'Негоденко Олена Василівна', and 'Парамонова Олена Вікторівна'. A 'Редагувати' button is at the bottom.

7

ВИСНОВКИ

Під час розробки інформаційної системи для надсилання військкоматом сповіщень призовникам було проведено аналіз вимог до інформаційної системи, виявлено та усунуто ряд недоліків на етапі макетування. Також систему було протестовано за допомогою методів ручного тестування та усунуто ряд недоліків, тому можна підсумувати що розроблена інформаційна система готова до впровадження.

Перспективи подальших досліджень

- Інтеграція з іншими месенджерами.
- Вдосконалення UI/UX зі збереженням швидкодії інформаційної системи.

8