

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Факультет математики і інформатики

Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

магістр

на тему “Розробка веб додатку для інформування населення щодо
відключення електроенергії”

Виконав: студент 2 курсу, групи МФ-61
спеціальність 122 «Комп’ютерні науки»
освітньо-наукова програма «Інформатика»

Бузулук Р. Р.

(прізвище та ініціали)

Керівник Бережная Н. І.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Харків – 2024 року

Зміст

1. Проблематика
2. Мета роботи

3. Архітектура додатку
4. Пінг роутера
5. Діаграма БД
6. Реєстрація
7. Дизайн
8. Технології
9. Віджет

- 10.Результат
- 11.Висновки

Проблематика

24 лютого 2022 року Росія вторглась в Україну. Країна агресор скоїла велику кількість військових злочинів, завдавала удари по цивільній інфраструктурі міст, знищувала атомні електростанції та інше. В осени того ж року населення України почало відчувати нестачу електроенергії, це призвело до планових відключень світла, вони дали змогу розподілити енергію приблизно рівномірно між будинками та містами. Людям довелось навчитись зберігати енергію, встигати



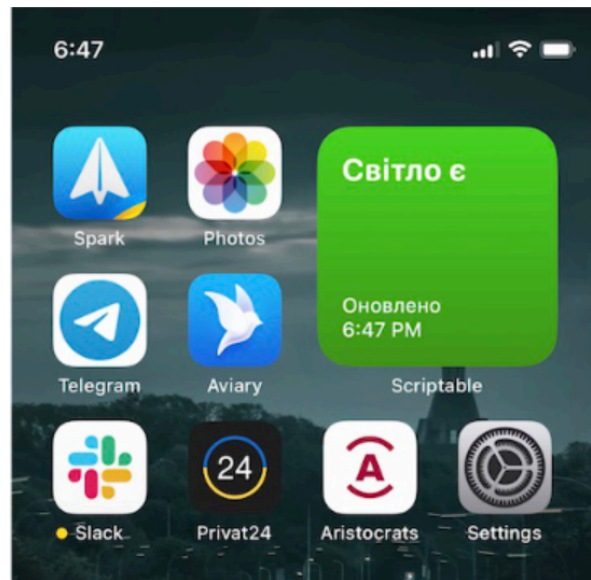
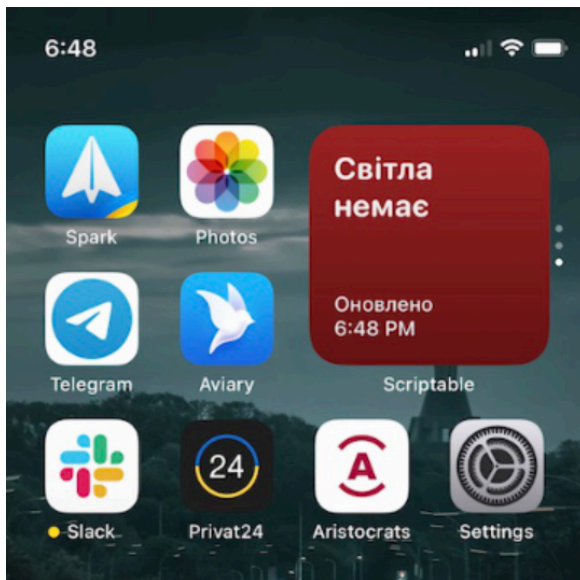
робити всі справи в короткий проміжок часу. Українське суспільство поборолو проблеми з електрикою завдяки віри, підтримки одне одно та українським підприємствам/компаніям які прийшли на допомогу громадянам. Для зручного планування часу, компанію “YASNO” створила таблиці в яких вводючи будинок можна дізнатись розклад за який в певному домі буде відбуватись відключення.

	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
Понеділок							✘	✘	✘	✘						✘	✘	✘	✘					
Вівторок	✘	✘	✘	✘						✘	✘	✘	✘						✘	✘	✘	✘		
Середа				✘	✘	✘	✘					✘	✘	✘	✘							✘	✘	✘
Четвер	✘						✘	✘	✘	✘						✘	✘	✘	✘					
П'ятниця	✘	✘	✘	✘						✘	✘	✘	✘						✘	✘	✘	✘		
Субота				✘	✘	✘	✘					✘	✘	✘	✘							✘	✘	✘
Неділя	✘						✘	✘	✘	✘						✘	✘	✘	✘					

Проблемою данного графіку була неточність. Іноді через чергові російські обстріли, іноді через непередбачувані поломки у системі, графік давав неправильну інформацію.

Виникла потреба в отриманні більш точної інформації. Було придумано багато ботів, сайтів та методів які могли в реальному часі дати

точну інформацію. В даній роботі я опиралась на вже існуючу розробку. Український айтивець Вадим Клименко створив зручний віджет за допомогою якого можна швидко перевірити наявність світла вдома. Але проблема полягає в тому що встановити цей віджет може лише користувач iOS. Також це система буде малокорисною за потреби перевірки декілька адрес. Тож метою даної роботи буде створення додатку який може покращити вже існуючі технології.

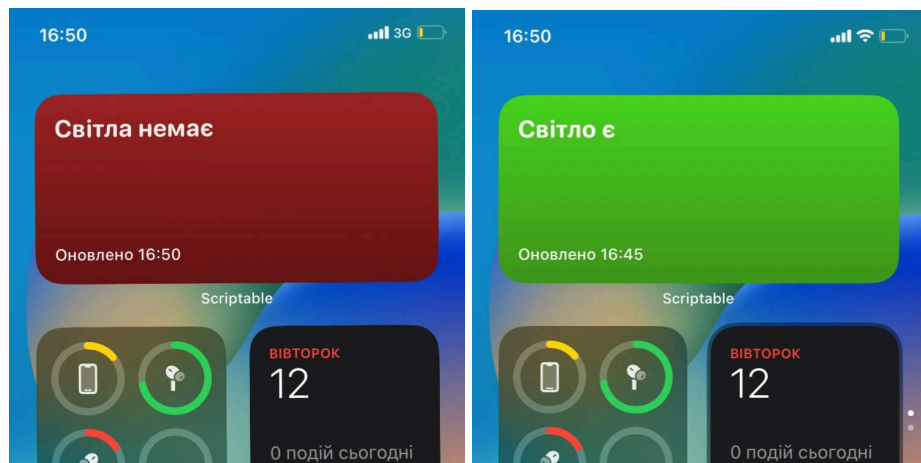


Мета роботи

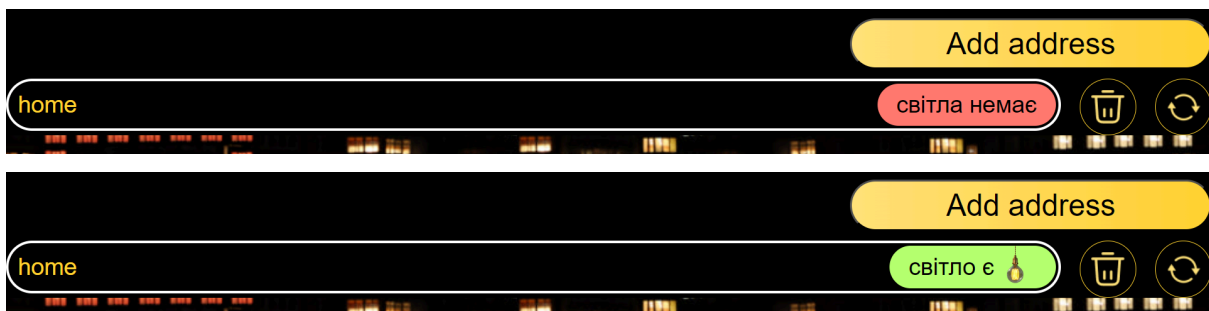
Додаток “Light in your home”- створений для інформування користувачів про наявність світла. Він умовно розділений на 3 частини Home, Віджет, Статистика.

Кожна з них дає інформацію про світло в вашій квартирі.

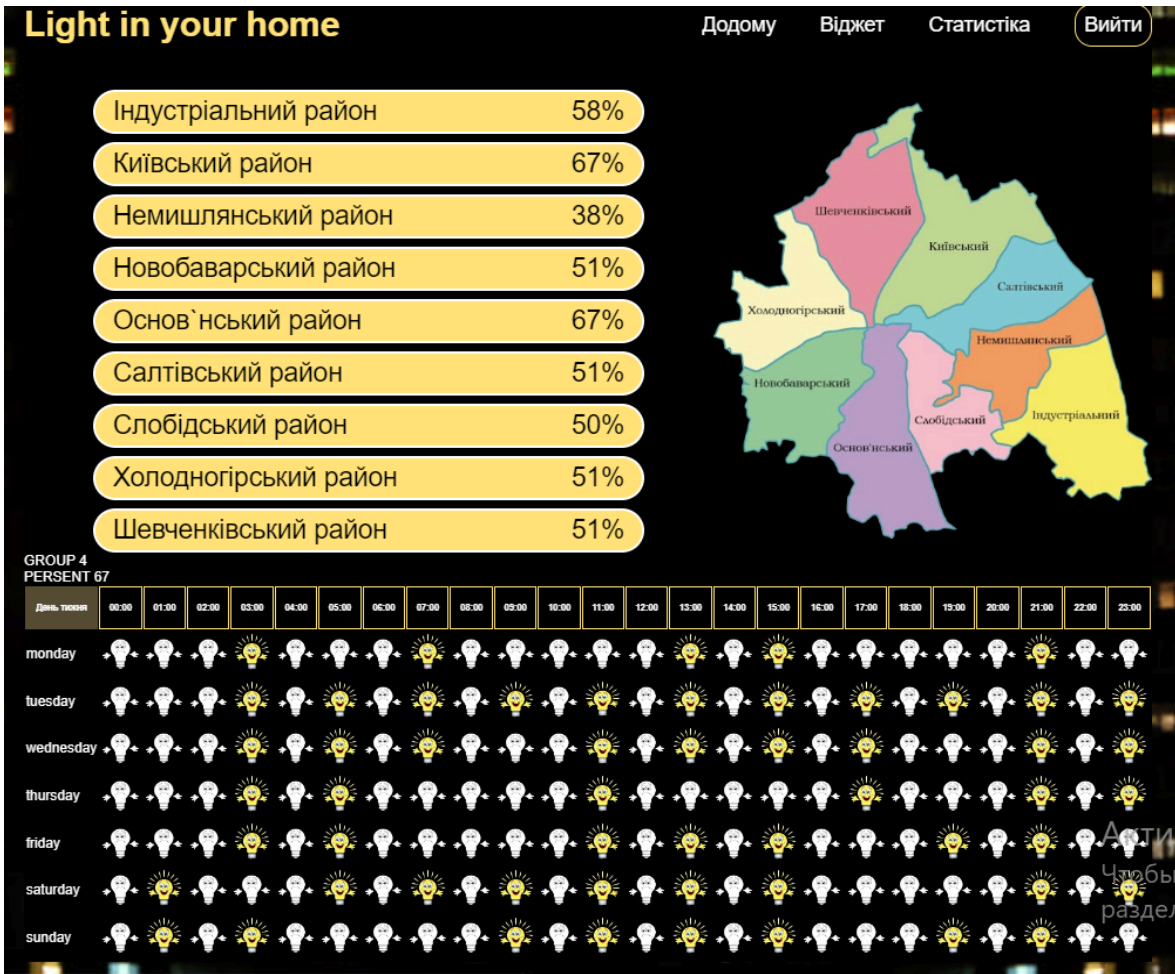
Віджет-надає інструкцію по встановленню на телефон(Iphone) віджету який може проінформувати вас про наявність світла. З недоліків - цей віджет працює тільки з айфонами, та кількість контрольних адрес обмежена однією.



Home - це головна сторінка додатку, вона працює за принципом віджета, але не прив'язана до моделі телефону, та не має обмежень по кількості контролюємих адрес, користувач може створювати будь яку кількість контролюючих блоків.



Статистика - це розділ який надає розклад відключені світла на тиждень.

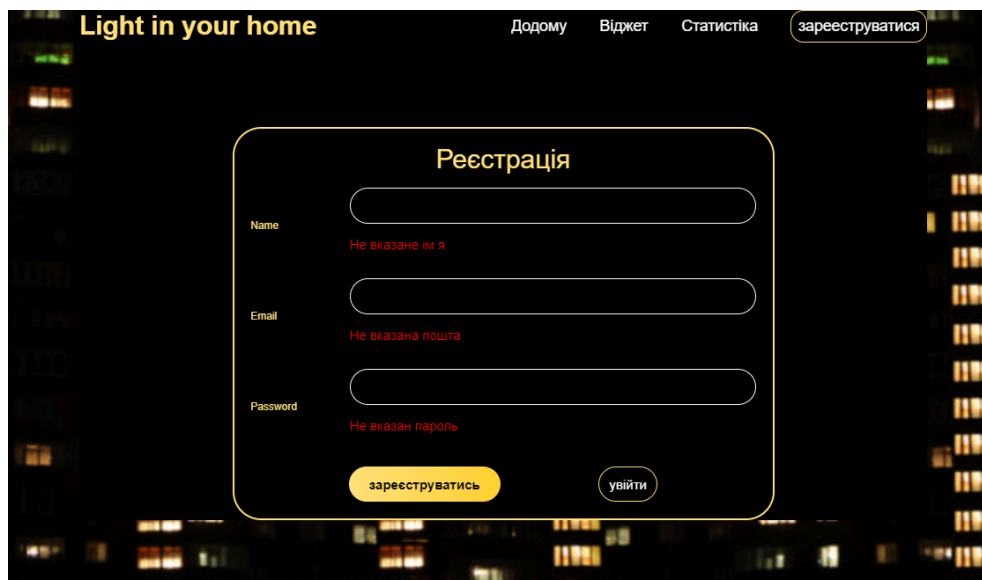


Реєстрація/авторизація - для того щоб мати можливість додавати свої адреси, та отримувати результат, користувач має бути авторизованим, переглядання інструкцій до віджету або статистика є доступними без реєстрації.

Архітектура додатку

Реєстрація

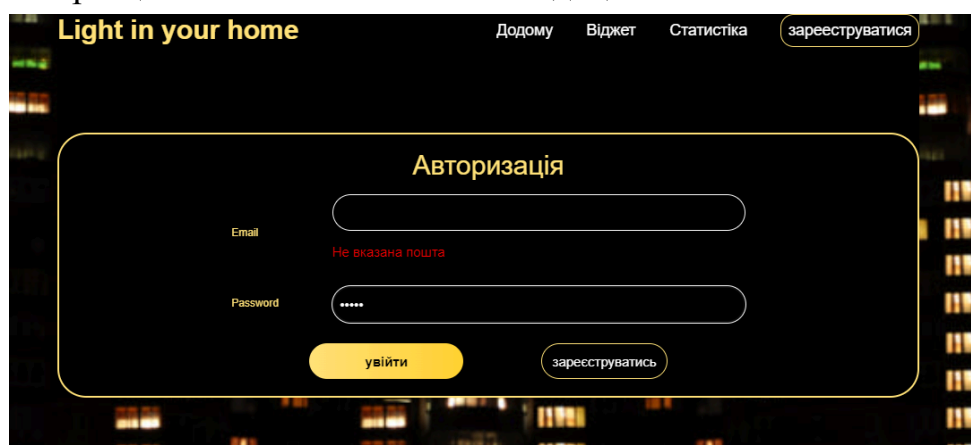
Сторінка реєстрації- це форма яка очікує від користувача дані про ім'я, email та пароль. На випадок некоректно введених даних на сторінці налаштована валідація.



The screenshot shows the registration form titled "Реєстрація" (Registration) within the "Light in your home" application. The form is set against a dark background with a city skyline at night. It features three input fields: "Name", "Email", and "Password". Each field has a red error message below it: "Не вказане ім'я" (Name not specified), "Не вказана пошта" (Email not specified), and "Не вказан пароль" (Password not specified). At the bottom of the form, there are two buttons: a yellow "zareestruvatisya" (register) button and a grey "uvityti" (login) button. The top navigation bar includes links for "Додому" (Home), "Віджет" (Widget), "Статистика" (Statistics), and a "zareestruvatisya" button.

Авторизація

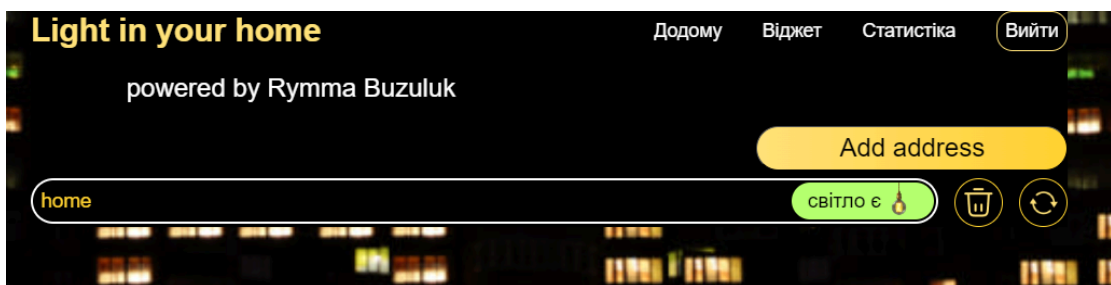
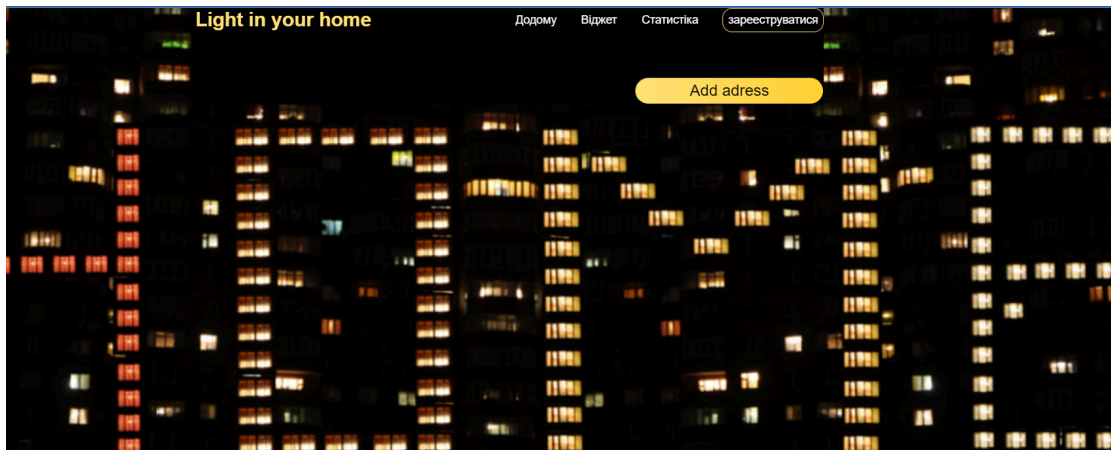
Сторінка авторизації - є подібною на сторінку реєстрації але ця форма запитує у вже зареєстрованих користувачів email та пароль. На цій сторінці також налаштована волидація.



The screenshot shows the authorization form titled "Авторизація" (Authorization) within the "Light in your home" application. The form is set against a dark background with a city skyline at night. It features two input fields: "Email" and "Password". The "Email" field has a red error message below it: "Не вказана пошта" (Email not specified). The "Password" field is masked with six dots. At the bottom of the form, there are two buttons: a yellow "uvityti" (login) button and a grey "zareestruvatisya" (register) button. The top navigation bar includes links for "Додому" (Home), "Віджет" (Widget), "Статистика" (Statistics), and a "zareestruvatisya" button.

Головна сторінка

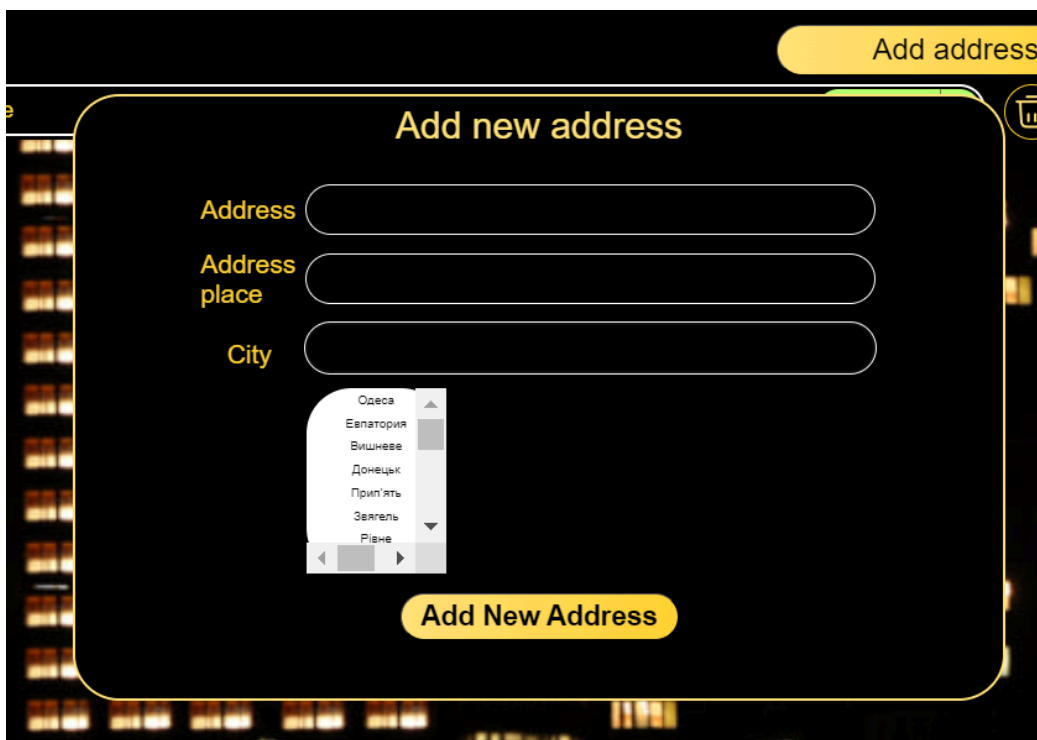
Головна сторінка додатку - не містить в собі ніякої інформації для незареєстрованого користувача. У випадку якщо користувач є зареєстрованим вона показує всі створені ним адреси. Також головна сторінка має кнопку "add address", ця кнопка додає можливість додати нову IP-адресу роутера.



Форма “Add new address” - створена для додавання нових компонентів які будуть демонструвати наявність світла за вказаною адресою. В цій формі користувач має вказати.

Address - ip адрес роутера.

Address place - це поле створене для зручності користувача, в ньому користувач дає назву місця в якому



розташована дана адреса(наприклад: офіс, дім, гараж...).

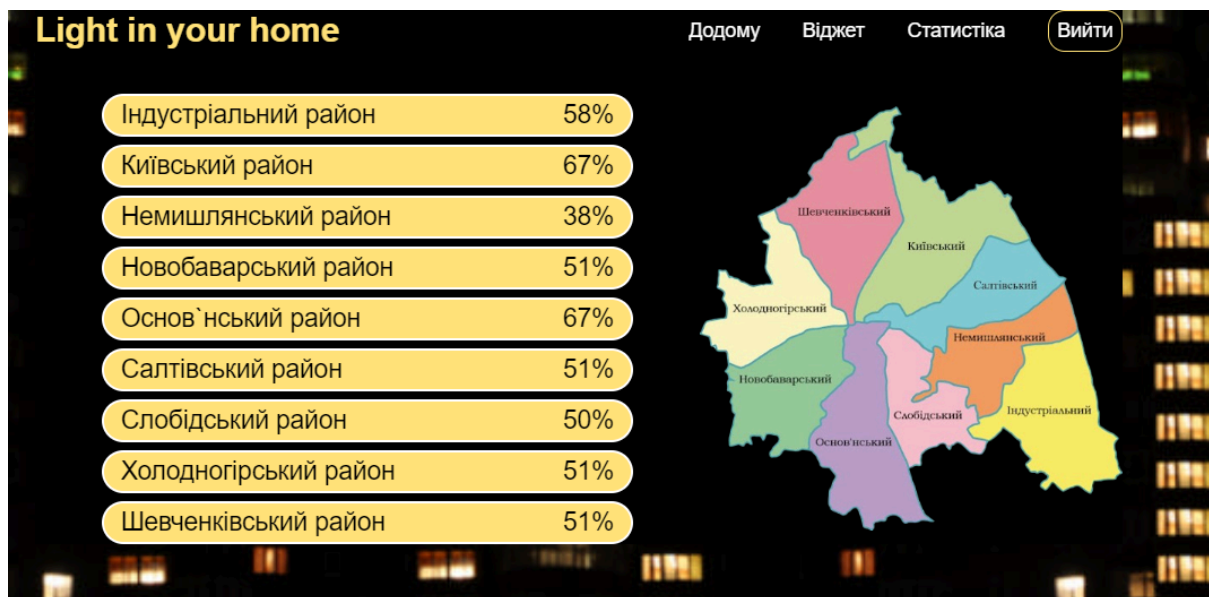
City - цей інпут створений для того щоб користувач міг сказати в якому місті знаходиться перевіряємий роутер. Ця інформація може бути корисною для майбутнього збору статистики.

Сторінка “Віджет”

Головна мета сторінка “Віджет” - дати користувачу інструкцію по встановленню віджета на телефон.

Сторінка “Статистика”

Статистика - це сторінка яка дає статистику на текущий тиждень, вона зроблена за прикладом графіку що надає компанія “yasno”. Сторінка розділена на дві частини, перша частина надає список районів харкова та відсоток нестачі електроенергії.



Друга частина сторінки це графік поточного тижня.

GROUP 1
PERSENT 51

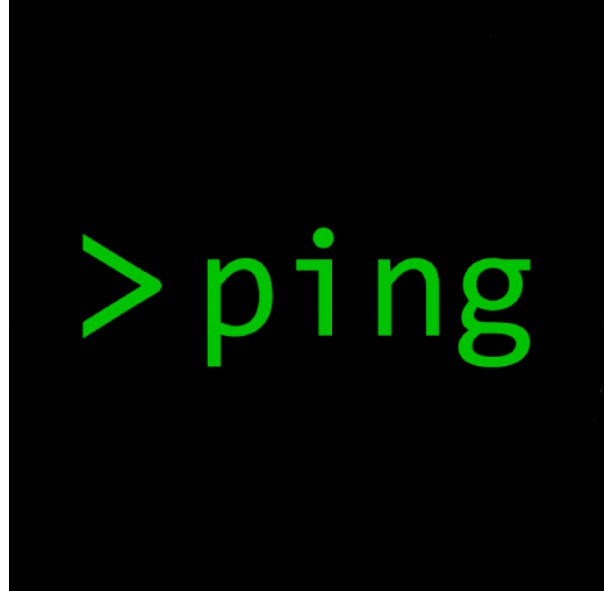


Аудио
Цитаты
разде

Пінг роутера

В даній роботі я використовую пінгування роутера для перевірки наявності світла в квартирі. Головними недоліками цього способу є те, що у разі відключення інтернету додаток буде давати результат “немає світла”, також проблемою може стати відсутність статичної IP-адреси, але її можна вирішити звязавшись з провайдером.

Пінгування роутера — це процес відправлення мережевих запитів, відомих як ICMP echo-запити, до роутера з метою перевірки його доступності та оцінки часу, необхідного для надсилання та отримання даних від нього. Пінг (ping) - це утиліта командного рядка, що широко використовується для виконання цього завдання.



Як відбувається пінгування роутера:

Надсилання пакетів: Коли ви відправляєте пінг до роутера, ваш комп'ютер або пристрій надсилає пакети даних на IP-адресу роутера.
Пересилання пакетів: Роутер отримує ці пакети та пересилає їх на адресу, вказану в пакеті.

Відповідь роутера: Роутер відповідає на пакети, відправляючи їх назад на ваш пристрій.

Вимірювання часу: Час, витрачений на відправлення та отримання пакетів, вимірюється та називається пінгом. Цей час вимірюється в мілісекундах і показує затримку між вашим пристроєм та роутером.

Аналіз відповідей: За відповідями від роутера можна оцінити якість та стабільність підключення до мережі. Великі затримки або втрати пакетів можуть свідчити про проблеми мережі.

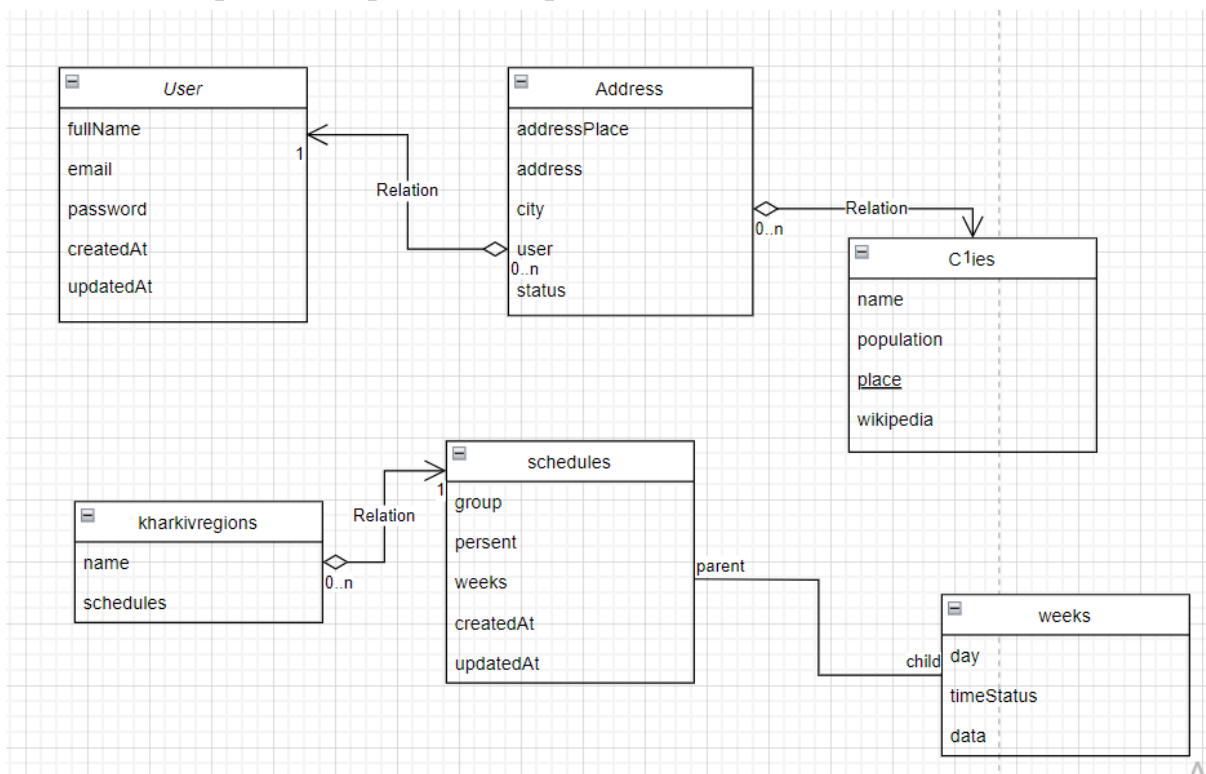
Пінг роутера може бути корисним інструментом для діагностики мережевих проблем, таких як низька швидкість передачі даних, високі



затримки або втрата пакетів. Воно також може використовуватися для моніторингу якості мережного з'єднання реального часу.

Діаграма БД

Основна мета створення Діаграми Баз Даних (БД) - це візуальне представлення структури бази даних та зв'язків між її сутностями. В проєкті використовується база даних, MongoDB, її особливість в тому що вона зберігає дані в форматі json. Немає принципової різниці в виборі бази даних, але мій вибір пав саме на MongoDB, бо на мою думку вона є зручною, легкою в використанні та повністю покриває потреби даної роботи.



Опис таблиці і полів:

User - це таблиця яка зберігає інформацію про користувачів додатку. Особливо важлива для реєстрації і авторизації користувачів.

Address - таблиця яка зберігає в собі інформацію про певну адресу користувача. Адресою є певний роутер який пингується та надає інформацію стосовно наявності світла (як це було описано раніше). Адреса обов'язково пов'язана користувачем який її створив.

City - це таблиця яка містить у собі всі населені пункти України, при створенні адреси до неї додається місто в якому вона розположена(для майбутньої можливості збирання даних).

Kharkivregion - таблиця яка містить в собі назви всіх районів Харкова, вона використовується для статистики по районах.

Schedules - таблиця розкладу, певний регіон Харкова має свій розклад відключення світла, ця таблиця містить розклади які присвоюються до певних регіонів.

Weeks - таблиця з таможнями це допоміжна таблиця до таблиці Schedules, вона містить в собі значення (true / false).

Реєстрація/авторизація

Структура додатку передбачає можливість створення сторінок користувачів, що є важливою функцією для збереження адрес та особистої інформації. З цією метою необхідно реалізувати механізми реєстрації та авторизації.

Процес розробки реєстрації включає стандартне заповнення форми, під час якого дані, введені користувачем, зберігаються у базі даних. Важливою особливістю цього процесу є обов'язкове застосування шифрування для збереження паролів. Шифрування паролів є ключовим етапом у забезпеченні безпеки інтернет-додатків, оскільки воно сприяє захисту конфіденційності користувачів та запобігає можливості несанкціонованого доступу до їх особистих даних.

Для шифрування даних використовується JWT(JSON WEB TOKEN).



Токени - це спосіб аутентифікації користувачів у веб-додатках. Вони генеруються при успішній аутентифікації користувача і використовуються для ідентифікації та авторизації на сервері. Токени дозволяють уникнути зберігання паролів на клієнтському пристрої та забезпечують безпеку під час передачі даних між клієнтом і сервером.

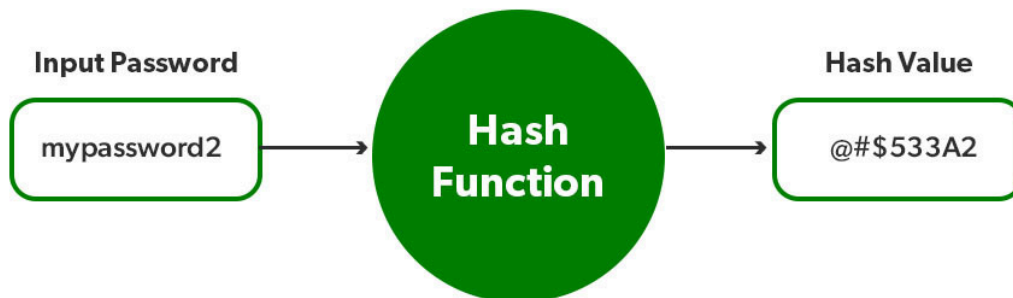
Реєстрація та отримання токенів: Під час реєстрації нового користувача йому присвоюється унікальний ідентифікатор, який зберігається у базі даних. Після успішної аутентифікації (зазвичай за допомогою електронної пошти та пароля), сервер генерує JWT (JSON Web Token) і відправляє його клієнту. Токен зазвичай містить інформацію про користувача та строк дії.

The image shows a registration form on a website. The form is titled "Реєстрація" and is set against a dark background with a blurred cityscape at night. The form contains three input fields: "Name", "Email", and "Password". Each field has a red error message below it: "Не вказане ім'я", "Не вказана пошта", and "Не вказан пароль". At the bottom of the form are two buttons: "zareestruvatis" and "uvyti". The website header includes "Light in your home" and navigation links: "Додому", "Віджет", "Статистика", and "zareestruvatis".

Використання токенів для авторизації: Після отримання токена клієнт зберігає його у локальному сховищі (наприклад, у localStorage або cookies). При кожному запиті до сервера клієнт надсилає токен разом із запитом. Сервер перевіряє токен, розшифровує його та перевіряє його цілісність та строк дії. Якщо токен валідний, сервер дозволяє доступ до захищених ресурсів.

Оновлення токенів та безпека: Токени можуть мати обмежений строк дії для забезпечення безпеки. Після закінчення строку дії токен потрібно оновити, щоб продовжити доступ користувача. Також важливо враховувати заходи безпеки, такі як захист від CSRF (Cross-Site Request Forgery) та XSS (Cross-Site Scripting) атак, шляхом належної обробки вхідних даних та використання HTTPS для захищеної передачі даних.

Бібліотека bcrypt допомагає гарантувати безпеку паролів та токенів шляхом захешування паролів перед їх збереженням, що унеможливорює прямий доступ до оригінальних паролів в разі витоку даних або несанкціонованого доступу до бази даних. Такий підхід допомагає забезпечити конфіденційність та цілісність даних користувачів в системі.



Дизайн

Додаток розроблений для користувачів які мають проблему з вимкненням світла, та має на мені надавати актуальну інформацію про відсутність/наявність електроенергії. Дизайн додатку відображає головну його тематику.

Тло додатку - це фото багатоповерхівки з вікнами зі світлом, головною метою при виборі тла було вибрати зображення яке передасть ідею та мету сайту.



Для створення дизайну додатку була використана платформа Figma - цей додаток використовується для розробки візуальної частини сайту. Зазвичай з цією платформою

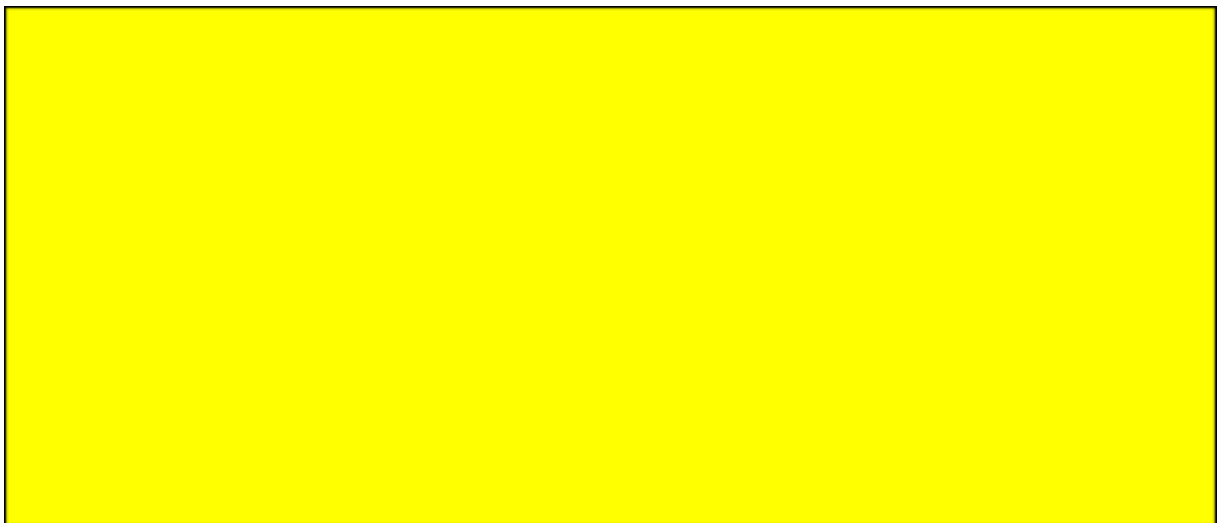


працюють дизайнери які презентуються зовнішній вигляд сайту, його елементи, сторінки і тд. Після вдалої презентації і згоди замовника дизайн передається розробникам які розпочинають роботу над створенням проекту. Але завдяки інтуїтивно зрозумілим інструментам скористатись платформаю може будь яка людина що прагне створити макет майбутньої програми.

Кольори додатку

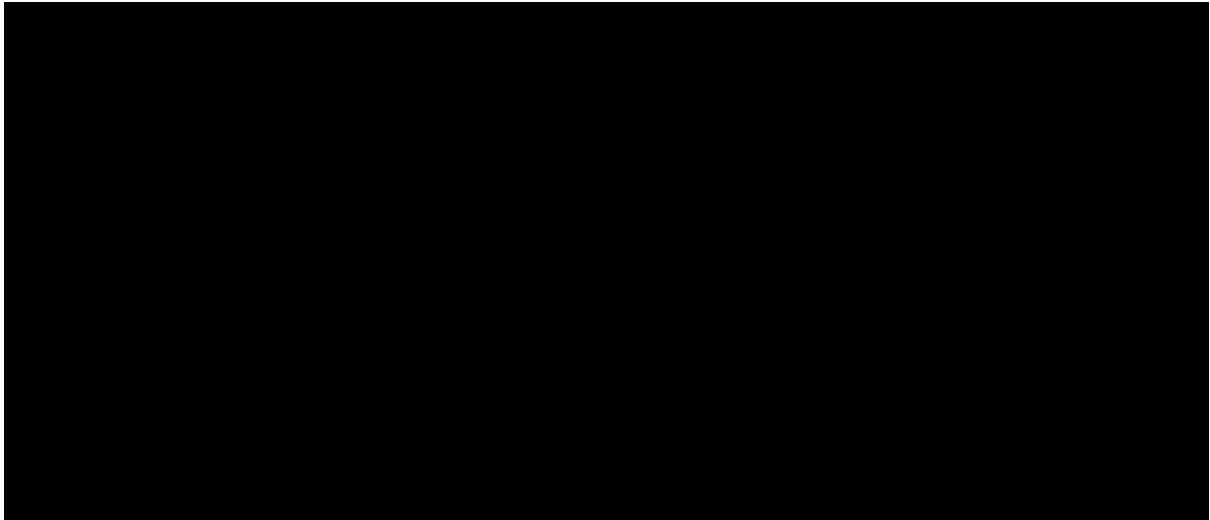
#ffe27a (жовтий)

Жовтий колір - є головним кольором додатку, він є своєрідною метафорою, що символізує освітлення та електрику , інтуїтивно асоціюється з теплом і затишком, він чудово відображає значення додатку - контролю за освітленням у будинку.



#000(чорний)

Чорний колір - на відміну від жовтого навпаки, у цьому додатку є

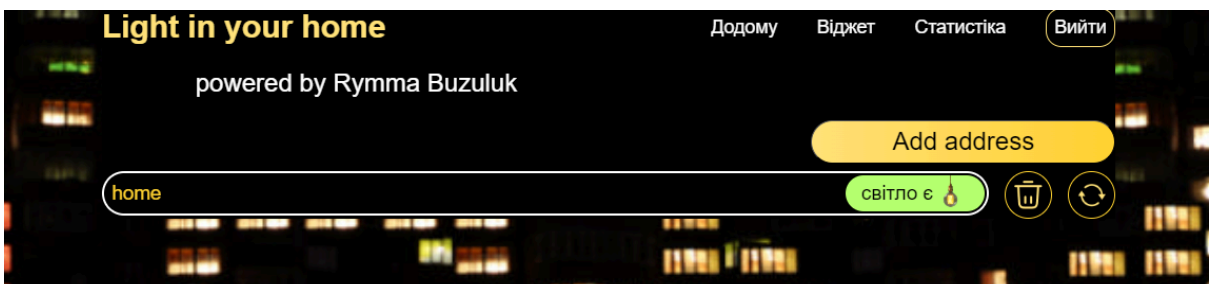


СИМВОЛОМ темряви.

#fff(білий)



Шрифт - Gilroy



Технології

Додаток містить в собі три частини frontEnd, backEnd, БД.

FrontEnd- його основа написана за допомогою бібліотеки React.js.



React-router-dom — це бібліотека яка використовується з React для маршрутизації веб-додатку.

Redux - це бібліотека що управляє станом застосунку у JavaScript-додатках, особливо в додатках, заснованих на бібліотеці React. Redux допомагає керувати станом(state) додатка, зберігаючи всі дані додатка в одному глобальному об'єкті, який називається "стор".



axios - це бібліотека, яка дозволяє здійснювати HTTP-запити з клієнтської частини вашого React додатку. У контексті React, axios використовується для отримання або відправки даних до сервера, обробки відповідей та оновлення стану компонентів з отриманими даними.

BackEnd- написана за допомогою Node.js.

Node.js - це платформа за допомогою якої створюються додатки, на мові JavaScript. Вона є досить розповсюдженою, бо розробник

express-validator - це інструмент для перевірки та валідації даних у веб-додатках, які використовують фреймворк Express.js в середовищі Node.js

Express - це мінімалістичний та гнучкий веб-фреймворк для Node.js, який дозволяє швидко створювати веб-додатки та API

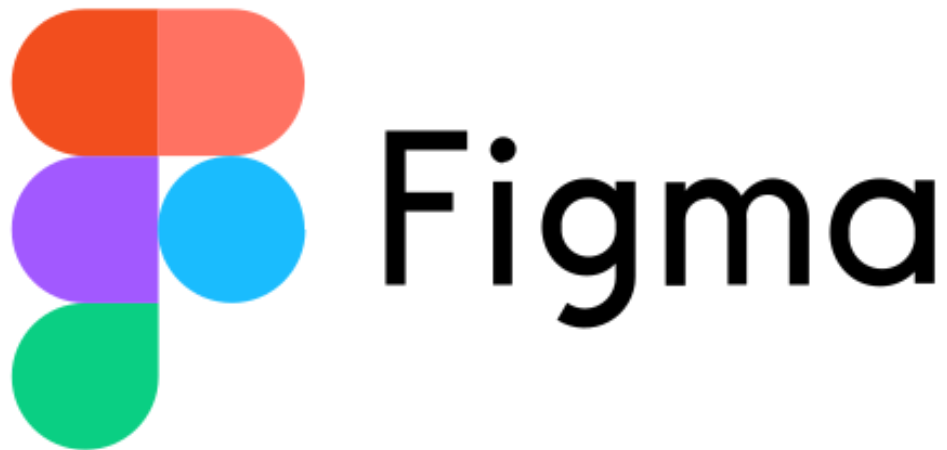
mongoose - це бібліотека для моделювання об'єктів даних (ODM) для MongoDB.

База Даних(БД) - це сховище яке зберігає дані, та надає можливість використовувати ці дані для роботи додатку.
В проєкті використана бд MongoDB.



Дізайн

Для створення дизайну була використана платформа Figma - це інтерактивний інструмент інструмент для дизайну та



прототипування веб-інтерфейсів і мобільних додатків. Вона дозволяє дизайнерам створювати інтерфейси, працювати над спільними проектами в реальному часі, спілкуватися з командою та клієнтами, а також створювати прототипи для тестування взаємодії користувача з додатком чи веб-сайтом.

Visual Studio Code (VS Code) - це безкоштовний текстовий редактор, розроблений компанією Microsoft. Він призначений для програмістів та розробників програмного забезпечення і надає широкий спектр функцій, які полегшують процес написання коду.



Visual Studio Code

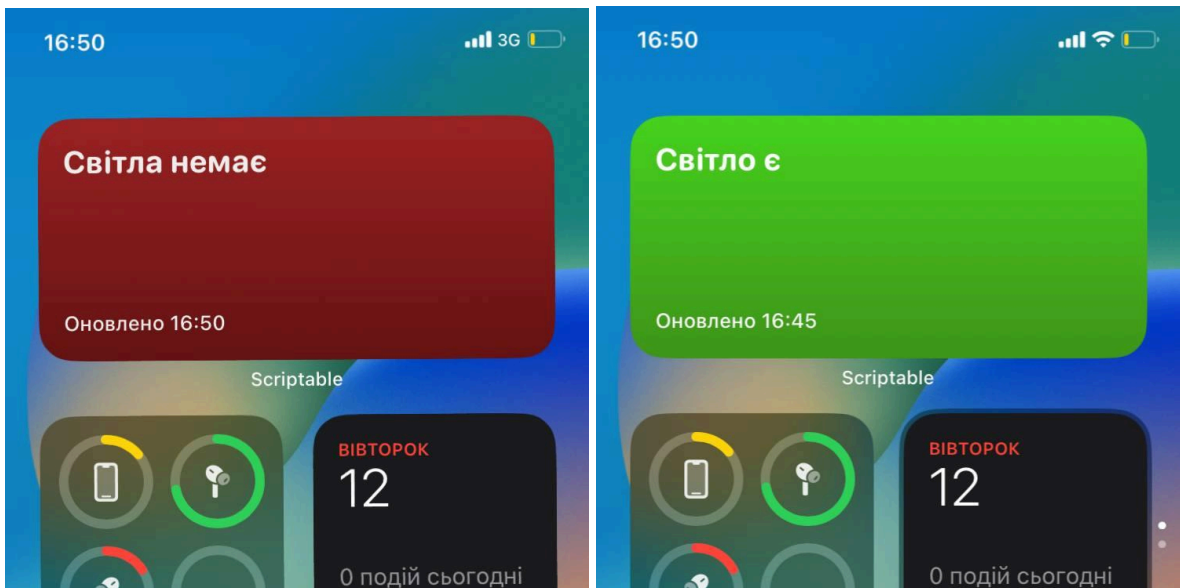
Віджет

Український розробник Вадим Клименко розробив віджет, який сповіщає про відсутність чи наявність вдома світла.

Про це він написав на форумі сайту DOU.

The image shows the word "DOU" in a very large, bold, black, sans-serif font. The letters are thick and have a slightly rounded appearance. The 'D' is on the left, the 'O' is in the middle, and the 'U' is on the right. The background is plain white.

Для того щоб розробити віджет, він використовував мобільний застосунок Scriptable, який призначений для створення віджетів на JavaScript. Принцип роботи простий: віджет перевіряє, чи доступний домашній роутер в інтернеті. Але автор зазначає що при відсутності інтернету та наявності світла віджет буде показувати червоний колір.



Автор зазначив декілька правил без яких віджет не буде працювати.

- 1) Вдома повинен бути завжди статичний IP.
- 2) Також iOS сама вирішує, коли їй оновити віджет, і робить це по документації в інтервалі 15..60 хвилин, але фактично виходить близько 15-20 хвилин. Якщо тапнути по віджету — він оновиться з актуальними даними.

Для того щоб встановити віджет собі на телефон, треба скористатись наступною інструкцією.

1. Повинен бути статический IP. Якщо роутер вимкнути-увімкнути, і IP зміниться — потрібно буде у провайдера замовляти додатково послугу статичного IP. Віджет постійно перевіряє лише одну і ту ж адресу.
2. Роутер повинен відповідати на ping запити. Якщо повертається помилка при натисканні на кнопку «Надіслати пінг на мій IP», то треба гуглити щось «%router name% Respond to Pings from WAN» та вмикати на роутері.
3. Потрібно встановити Scriptable, завантажити файл нижче, поділитись ним в Scriptable та замінити IP в першому рядку на потрібний. Раптом що, ось відео, як це робиться.
4. Створити віджет та додати його на домашній екран.
5. Всього!

Результат

Результатами даної роботи є веб додаток який призначений для користувачів які мають запит на перевірку електрики в своєму приміщенні. Додаток складається з 3 частин, кожна з яких описує різні способи отримання інформації.

- пінгування роутера
- таблиця відключень
- віджет

Кожна частина має свої переваги і недоліки які були описані вище, саме тому було важливо об'єднати їх у спільному проєкті для створення платформи яка може надати користувачу всю потрібну інформацію.

Висновки

В результаті роботи, вийшло створити web-застосунок для контролю наявності електроенергії з трьома способами отримання даних.

Була визначена актуальність теми, за відсутності ресурсу який міг би надавати повну інформацію щодо відключень, дана робота збирає декілька способів отримання інформації в один проект. Проект може бути допрацьований за допомогою розширення функцій збору статистичних даних, додаткової інформації щодо розташування пунктів незламності на мапі.

Проект був розроблений за допомогою сучасних технологій, використовувалась платформа VS Code, так як ця система має очевидні переваги, згадані раніше.

Проект пройшов етапи збору вимог, планування, проектування, розробки і тестування.

Також в ході практики було здійснено пошук і вивчення наукової літератури з обраної теми,— вивчено та проаналізовано методи вирішення наукових завдань з обраної теми, виконано розробку фрагментів конкретного проекту, реалізованого колективом працівників базового підприємства або інших студентів.

Такими чином, поставлені цілі і завдання були повністю досягнуті в рамках даної виробничої практики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 3 Д. Флаунт. "Розробка веб-інтерфейсів". М.: Олімп-Бізнес, 2010.
- 4 У. Ройс. "Управління проектами для створення програмного забезпечення". М.: Лорі, 2002.
- 5 Л. Константайн, Л. Локвуд. "Розробка програмного забезпечення". СПб.: Питер, 2004.
- 6 В. В. Липаєв. "Методи забезпечення якості крупномасштабних програмних засобів". М., Сінтег, 2003.
- 7 Леффінгуелл Д., Уїдріг Д. "Принципи роботи з вимогами до програмного забезпечення. Уніфікований підхід". Пер. з англ. - М.: Вільямс. 2002.
- 8 Липаєв В.В. "Системне проектування складних програмних засобів для інформаційних систем". Вид. друге перероблене і доповнене. - М.: Сінтег. 2002.
- 9 Соммервілла І. "Інженерія програмного забезпечення". Пер. з англ. - М.: Вільямс. 2002.
- 10 Якобсон А., Буч Г., Рамбо Дж. "Уніфікований процес розробки програмного забезпечення". Пер. з англ. - СПб. Пітер. 2002.
- 11 М. Макконнелл. "Розробка веб-інтерфейсів: Ідеї та патерни". М.: Лорі, 2005.
- 12 Д. Крамер. "Проектування користувацьких інтерфейсів веб-додатків". Нью-Йорк: О'Рейллі, 2009.