|  |  |
| --- | --- |
| **Project Title** | **Application Integration - API** |
| **Qualification Name (NICF)** | **Application Integration (API using Spring Boot & React JS)** |
| **Product Name** | **Applied Degree in Software Engineering** |
| **Module Name (NICF)** | **Application Integration (API using Spring Boot & React JS)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name** | | **Assessor name** | |
| Wildan Luqmanul Hakim | |  | |
| **Date issued** | **Completion date** | | **Submitted on** |
| 12 December 2022 | January 2023 | | January 2023 |
|  | |  | |
| **Project title** | **Application Integration - API** | | |

|  |
| --- |
| **Learner declaration** |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.    Student signature: Date: January 2023 |

1. Project Background

Spring Boot was used in the development of the Know-Your-Neighborhood app. The project's objective is to rewrite the application using React.Js and Spring Boot while utilizing the current APIs for logging in and out. An API should be available for your application's login button. The project's scope also includes examining and comparing various existing APIs sample by sample, evaluating each one's suitability, spotting any potential security issues, and implementing chosen APIs on live websites. a login to be created.

1. Project Objective

* Be able to design and develop a backend using Spring Boot and JPA Framework.
* Be able to develop API using Restful Web Services.
* Be able to develop frontend application using React JS.
* Be able to identify existing APIs and its uses in already developed application

You have already developed a "Know-Your-Neighborhood" application. The goal of this application is to provide login/sign up using existing API. For this to happen, the application should have login button with available APIs.

The Know-Your-Neighborhood website consists of the following Key pages

1. Home Page
2. Registration Page
3. Login Page with API link
4. Contact us Page
5. About us Page
6. Terms and Conditions Page

Customers can login using the existing API and fetch basic information such as name, email from API.

Tools & platform used

1. Microsoft Word
2. Microsoft power point
3. Diagrams.net
4. Spring Boot
5. Spring Initializr
6. React JS
7. Visual Studio Code
8. Postman
9. MySQL Workbench
10. Mozilla Firefox
11. Google Chrome
12. Facebook API
13. Project Requirement Specification

* Node JS (LTS Version)
* MySQL 8
* Java 11

1. Task 1

**APIs and Type of APIs**

1. Explain what API is, its role and need for API and research existing APIs.
2. Examine the relationship between API and SDK.
3. Identify types of API and its uses.
4. Identify the potential security issues with API and critically evaluate the suitable API for given scenario or your selected application.

**Solution:**

1. **Explain what API is, its role and need for API and research existing APIs.**
2. What is API?

Application Programming Interface is referred to as API. It is a system of guidelines, procedures, and building blocks for software and applications. An API enables communication between various systems and specifies how software components should cooperate. It serves as a middleman, enabling secure and controlled data and functionality sharing between various systems. Software programs or web-based services can implement APIs, which give developers access to the functionality or data of the service so they can create their own applications. This allows for greater flexibility and scalability in software development as well as the integration of various systems, services, and programs.

1. Its Role and Need for API

API is essential to software development because it enables secure and controlled data and functionality sharing and communication between various software systems and applications. The API serves as a middleman, offering a set of guidelines and protocols for how the various systems should communicate with one another. By utilizing the functionality of current systems, developers are now able to build new applications and services without having to start from scratch.

The increasing number of software programs and services that are accessible both online and within organizations has led to a need for APIs. As more and more systems and services are developed, it becomes increasingly important that they share information and functionality. Without APIs, different software systems would be unable to communicate with one another and share information, which would limit their usefulness and flexibility.

1. Example APIs for mobile, desktop, Web APIs.

There are many existing APIs available for various services such as social media, e-commerce, weather, financial data, and more. Some well-known APIs are as follows:

* Google Maps API - Enables programmers to access mapping information like satellite imagery and street maps and embed Google Maps into their own applications.
* Twitter API - Enables the creation of applications that can interact with the Twitter platform and the access to tweets and other Twitter data for analysis.
* Facebook API - enables the development of social media apps and services by giving developers access to user data and the ability to post data to the Facebook platform.

1. **Examine the relationship between API and SDK.**

A set of guidelines and procedures known as an API are used to create and integrate software applications. APIs enable communication between various systems while defining how software components should interact. They serve as middlemen, enabling secure and controlled data and functionality sharing between various systems.

On the other hand, an SDK is a set of resources and tools that programmers can use to create software applications. It is created to make it simpler for developers to work with a specific platform or technology and typically includes libraries, documentation, and sample code.

The relationship between API and SDK is that an SDK usually includes an API. SDKs are built on top of APIs and provide additional functionality and resources to developers. An SDK provides an API as a set of tools to interact with the underlying platform or technology, as well as additional resources such as libraries, sample code, and documentation. This makes it easier for developers to work with the platform or technology, as they have all the necessary tools and resources in one place.

1. **Identify types of API and its uses.**
2. Examine different APIs

There are several types of APIs, including:

* Open APIs, also referred to as external or public APIs, are freely accessible to developers and other users. They might demand logging in, using an API key, or using OAuth.
* Internal APIs, also referred to as partner APIs or private APIs, are employed within an organization. They could be used by staff members or by outside partners.
* Composite APIs are a combination of multiple endpoints and data sources that are combined into a single API.
* Webhooks are an API type that supports real-time communication and is frequently used for occasions like notifications or updates.

1. Examine the uses of APIs for a particular type.

Different software systems can communicate with one another thanks to APIs. They enable the development of new applications and services by allowing various systems to communicate and exchange data.

1. **Identify the potential security issues with API and critically evaluate the suitable API for given scenario or your selected application.**

API security is essential for safeguarding private information and guaranteeing the reliability of the systems that use APIs. The following are some possible security issues with APIs:

* Injection attacks: When an attacker is able to insert malicious code into an API in order to access or manipulate data, this is known as an injection attack. This includes script injection, which allows an attacker to run arbitrary scripts, or SQL injection, which allows an attacker to execute any SQL commands.
* Authentication and access control: Attackers might have access to private information or be able to carry out unauthorized tasks if APIs do not properly authenticate users and enforce access controls.
* Data leakage: Attackers may be able to intercept and read sensitive information if APIs do not properly encrypt data.
* Lack of rate limiting: Attackers might be able to launch a denial of service attack by flooding the API with requests if APIs do not restrict the rate at which requests can be made.
* Weak or compromised keys: APIs that rely on keys for authentication may be exposed if those keys are unreliable or come into breach.

The following factors should be taken into account when determining which API is best for a particular scenario or application:

* Security: To guard against potential vulnerabilities, the API needs to have strong security measures in place.
* Compatibility: The API should be able to work with the application's technology stack.
* Functionality: The API need to offer the features your application requires.
* Performance: The API should operate quickly, with minimal latency, and with a high degree of availability.
* Documentation: The API need to have thorough, simple-to-understand documentation.
* Support: The API should have good support from the vendor, including email support, forums, and tutorials.
* Pricing: The API should have a pricing structure that fits your needs and financial constraints.

1. Task 2

**Apply the knowledge of API research to design an application.**

1) Analyze the given scenario, identify the requirements and select the appropriate API.

2) Develop relevant wireframes for using the API for specific purposes.

3) Identify your scope and target platform.

4) Evaluate and justify your choice of APIs for your application. (Shows security for the selected API.)

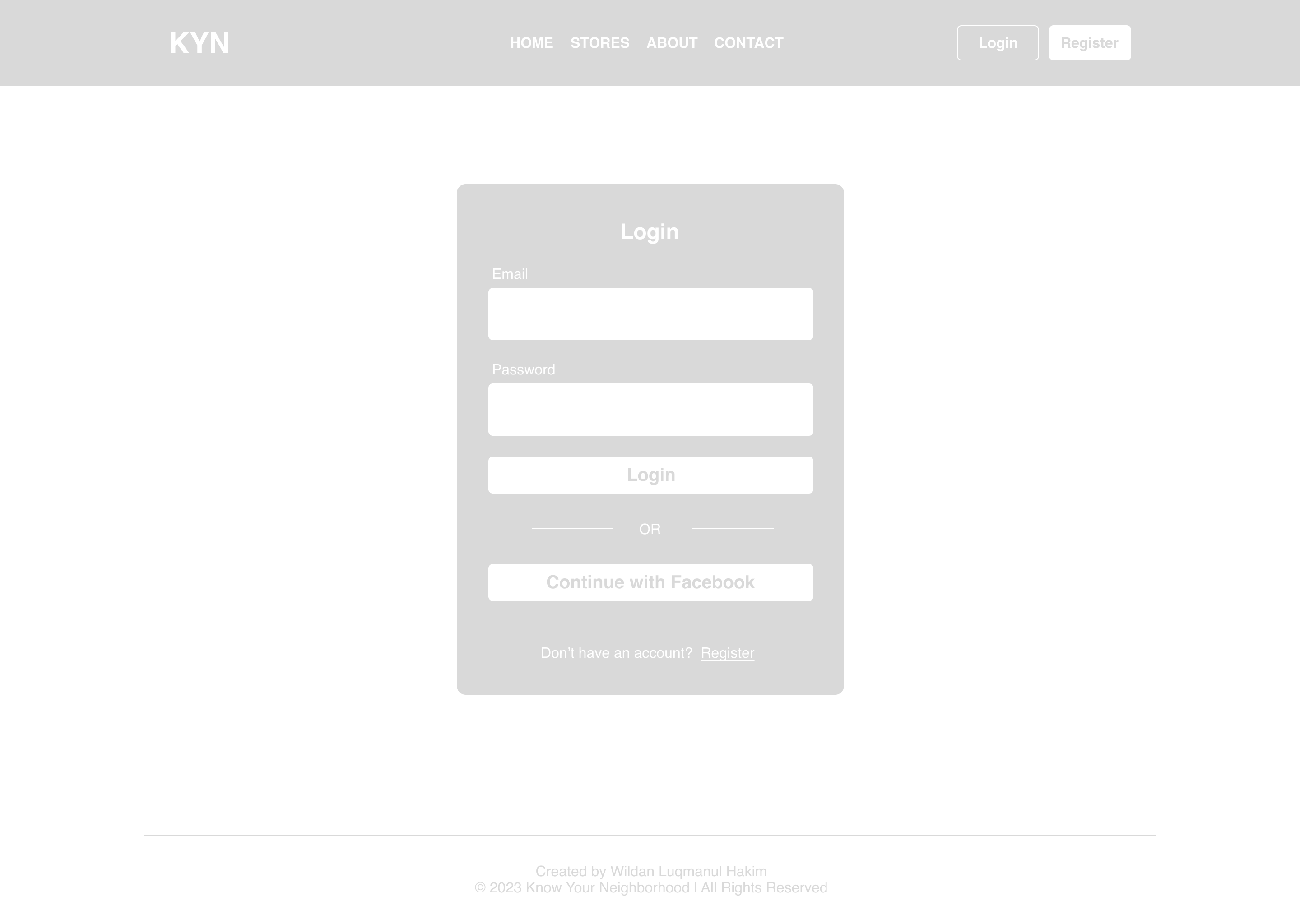
**Solution:**

1. **Analyze the given scenario, identify the requirements, and select the suitable API for the same.**

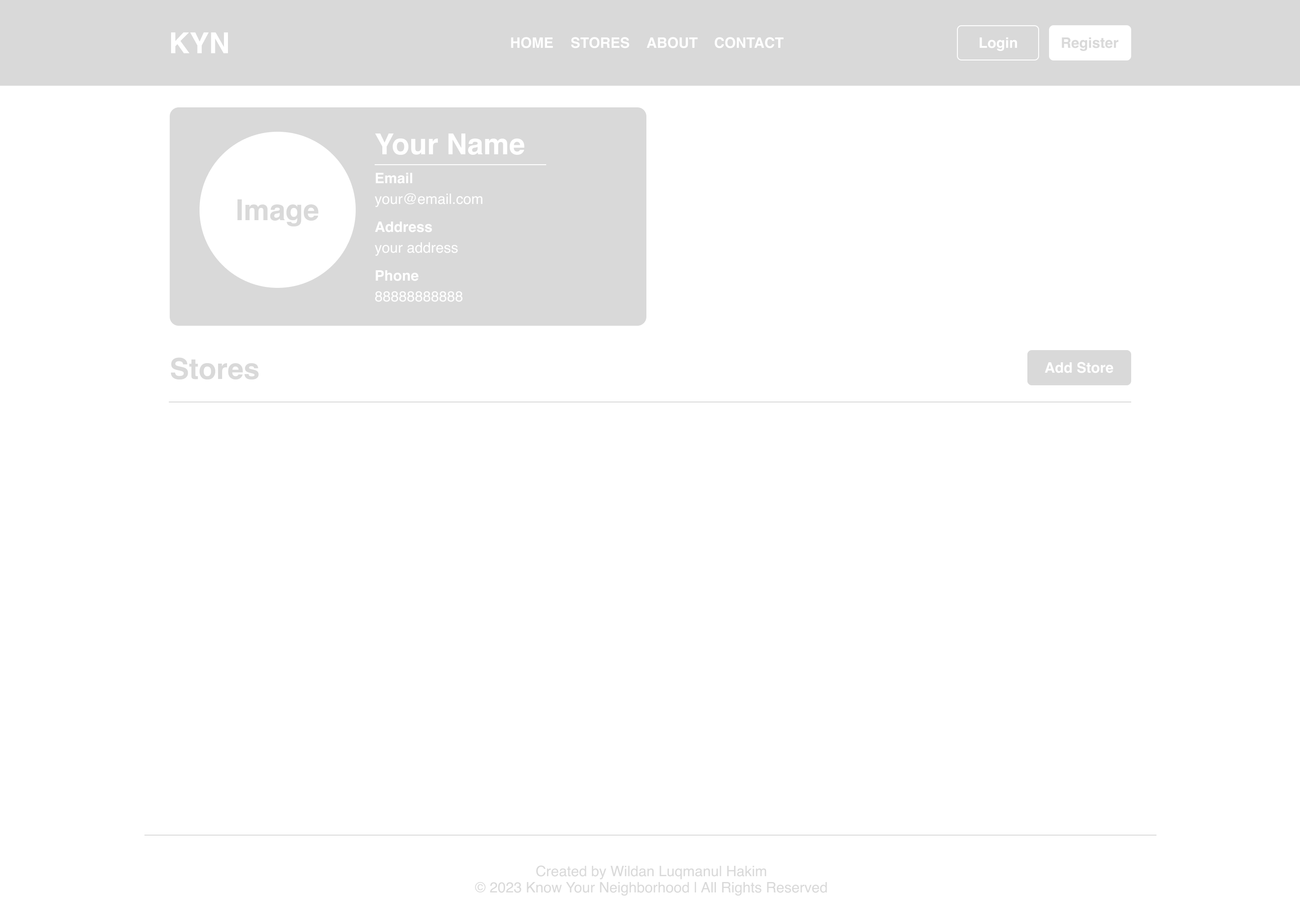
Facebook offers the Facebook Login API feature, which enables users to sign in to other websites and apps with their Facebook credentials. It is a method for website and app developers to use Facebook user data to authenticate and authorize users on their platforms.

Two scenarios are made possible by Facebook Login: authentication and requesting users' consent to access their data. Facebook Login can be used both for authentication and data access, or just for authentication. The user can access the Know Your Neighborhood website by logging in with their Facebook account, and the API will display and use their profile.

1. **Develop the relevant wireframes to utilize the API for given purpose.**
2. Login Page showing Login with Facebook Button and Form Input Login Wireframe



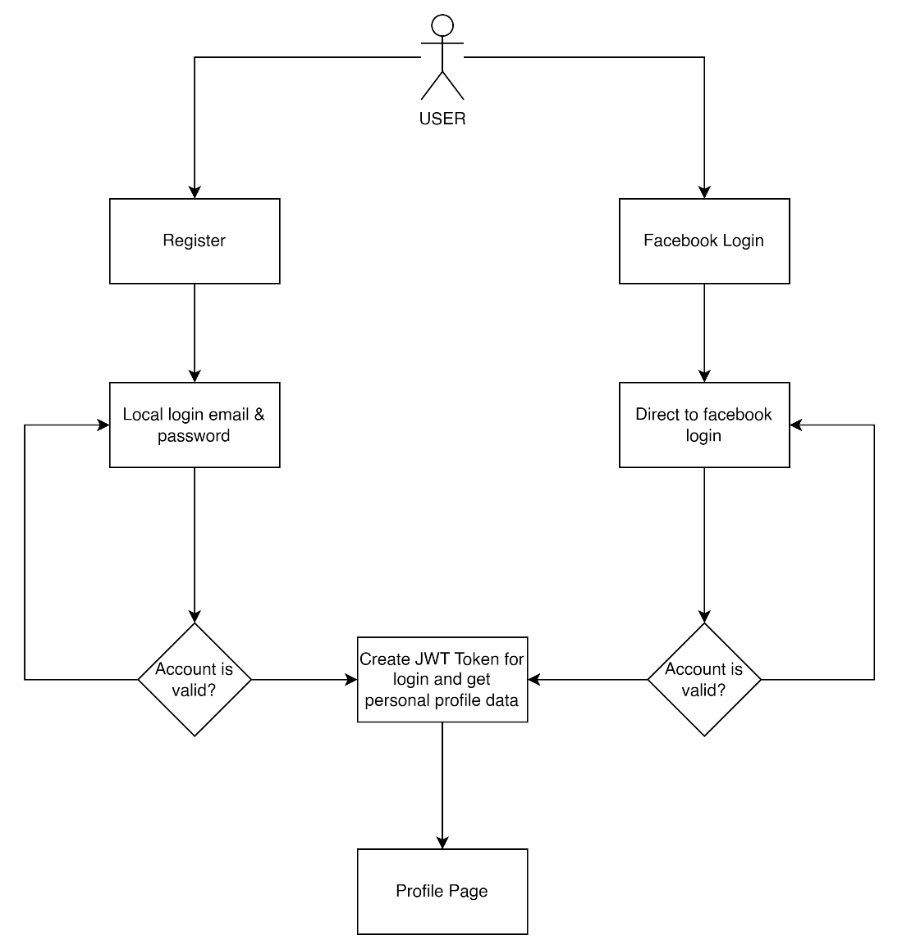
1. KYN dashboard Wireframe after Facebook login API



1. **Identify the scope and target platforms.**

The vast majority of devices and operating systems now support Facebook's APIs. It can, for instance, be used on desktop computers and mobile devices, as well as in Mac and Windows environments.

1. **Evaluate and justify the selection of chosen APIs for the application. (Show security of selected APIs.)**

****

The user has the option of using the website's local login feature or their Facebook login to access the website. If a user chooses a local login, they can register first, log in using their email address and password, and then are directed to their profile page once they have successfully logged in. When a user selects Facebook Login, they are taken to the Facebook Login page where they must enter their credentials. Once they have successfully logged in, they are then taken to their profile page.

**Security of selected API for the application**

* **OAuth**

OAuth is used as an open authorization protocol to give third-party applications working on behalf of resource owners restricted access to HTTP services. It can accomplish this without revealing the user's identity or permanent credentials. Additionally, a third-party application may make use of it on its behalf. According to the OAuth working principal, user authentication is delegated to a service that hosts a user's account, and third-party applications are given access to the user's account.

1. Task 3

**Application Implementation**

1. Introduce three different types of backend, frontend, and API implementation process
2. Discuss a range of suitable development environments for front-end and back-end to develop an application
3. Develop a backend and Web service using selected development environment for given scenario
4. Develop an application that utilizes an API.
5. Construct the application which implements the selected API in Task 2.

**Solution:**

1. **Introduce three different types of backend, frontend, and API implementation process**
2. Three different types of Backend

* Express JS
* Spring Boot
* Laravel

1. Three different of frontend

* React JS
* Angular JS
* Vue JS

1. Three different of API

* REST API
* OPEN API
* RPC API

1. **Discuss a range of suitable development environments for front-end and back-end to develop an application**

Front-end development environments:

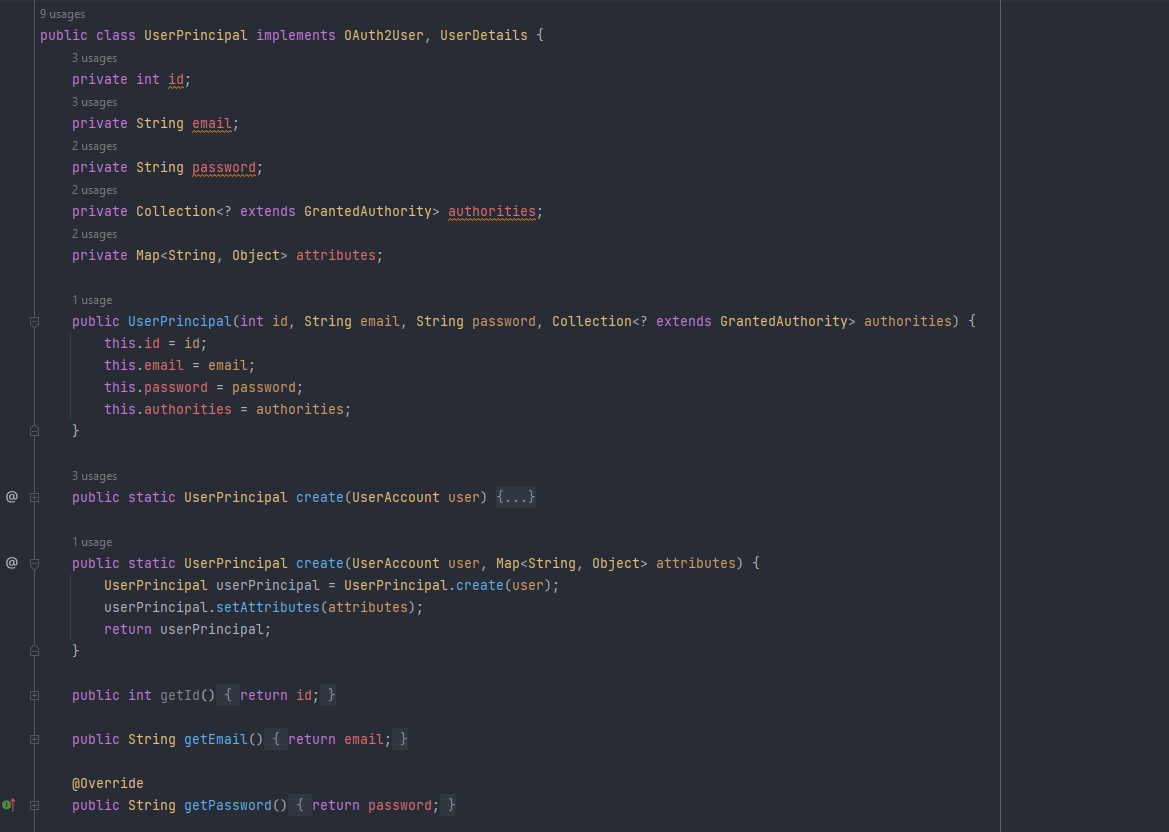
* Visual Studio Code: a well-liked open-source code editor that works with Git and a variety of programming languages and has built-in debugging.
* Sublime Text: a robust, portable text editor with a simple interface, a variety of plugins, and customizability options.
* WebStorm: a potent integrated development environment (IDE) for front-end technologies like TypeScript and JavaScript.

Back-end development environments:

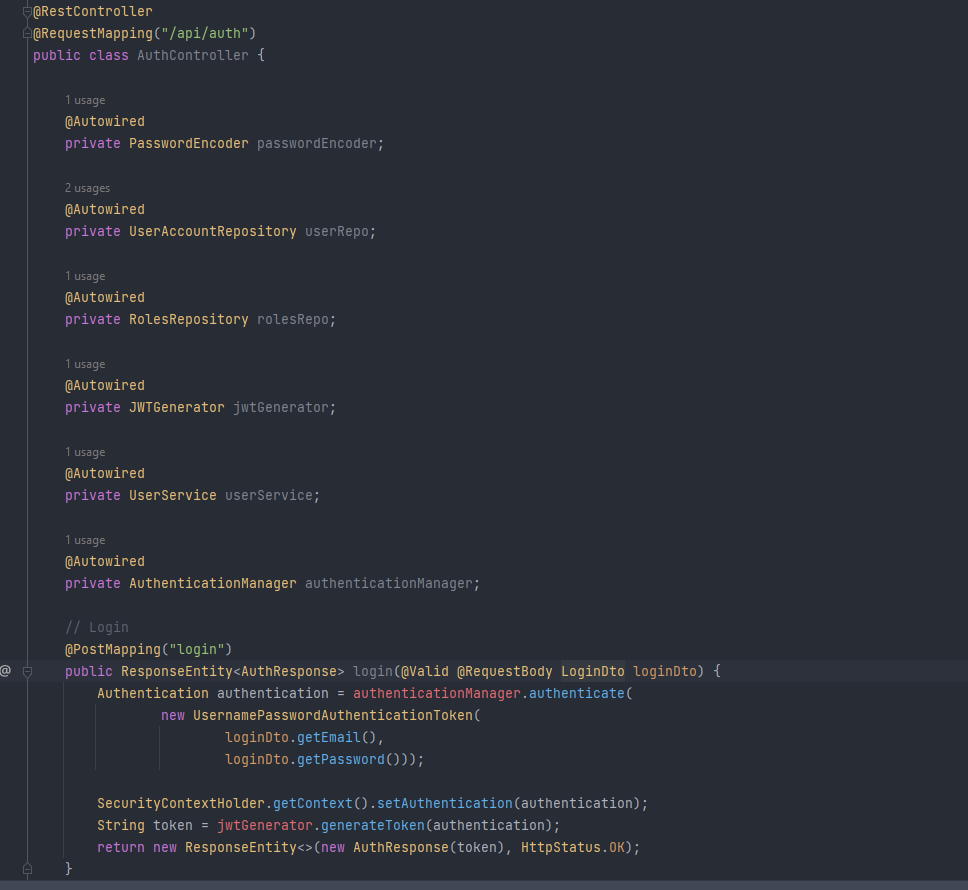
* Visual Studio: Microsoft created a well-known, feature-rich IDE that supports a large number of programming languages and frameworks..
* Eclipse: an IDE that is widely used and is open-source and supports a variety of programming languages and frameworks.
* IntelliJ IDEA: a well-known IDE for Java development that is fully featured and supports a variety of other programming languages and frameworks.

1. **Develop a backend and Web service using selected development environment for given scenario**

* **Backend Local Login**

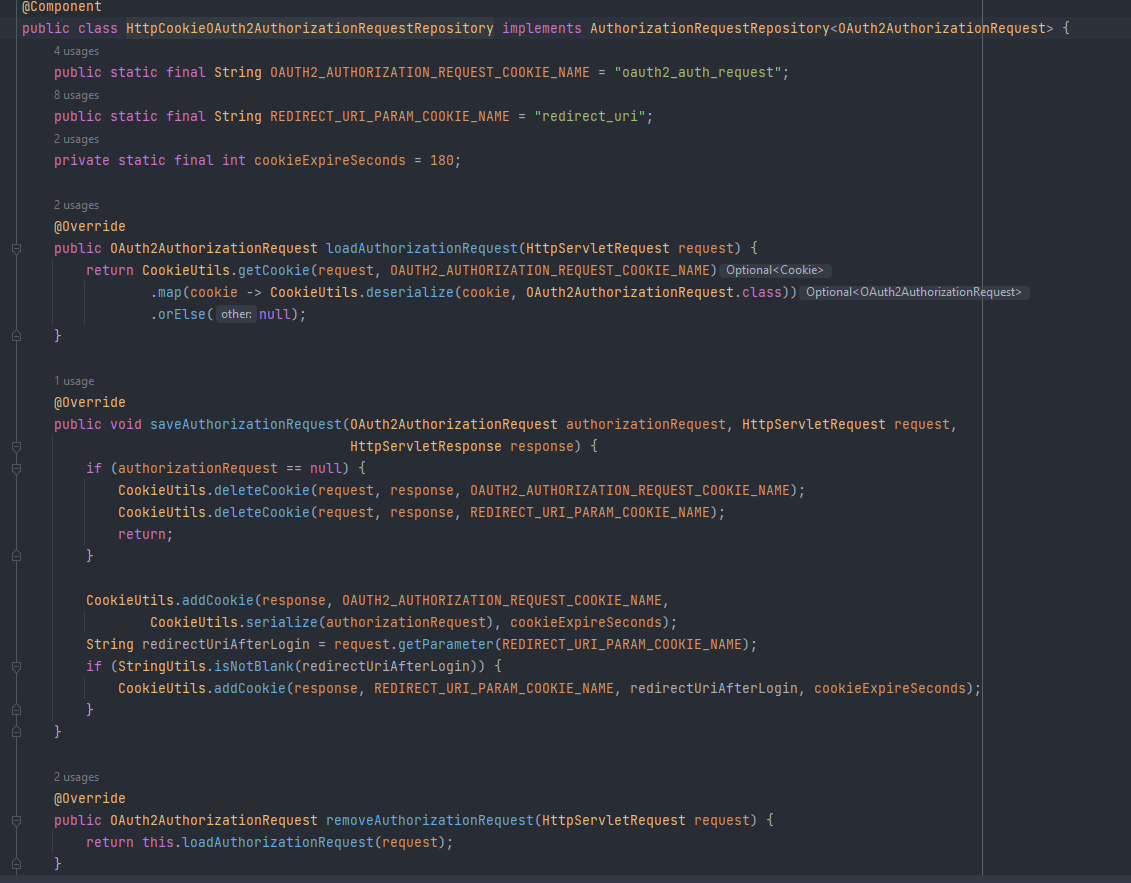
UserPrincipal.java  


UserDetailsServiceImpl.java  


AuthController.java  


* Backend Facebook Login

FacebookOAuth2UserInfo.java  


HttpCookieOAuth2AuthorizationRequestRepository.java  


1. **Develop an application that utilizes an API.**

* **Backend Local Login**

auth-context.js

****

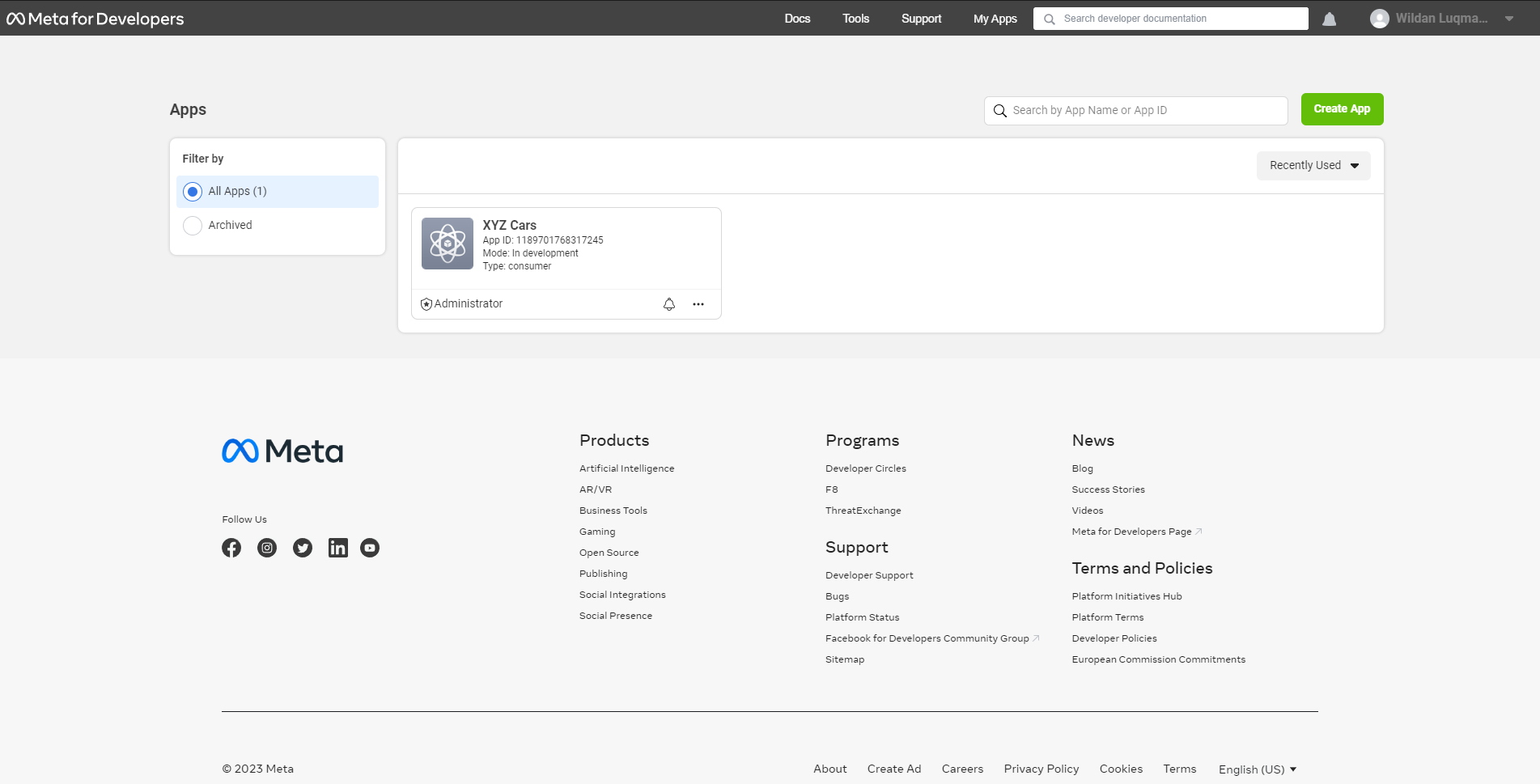
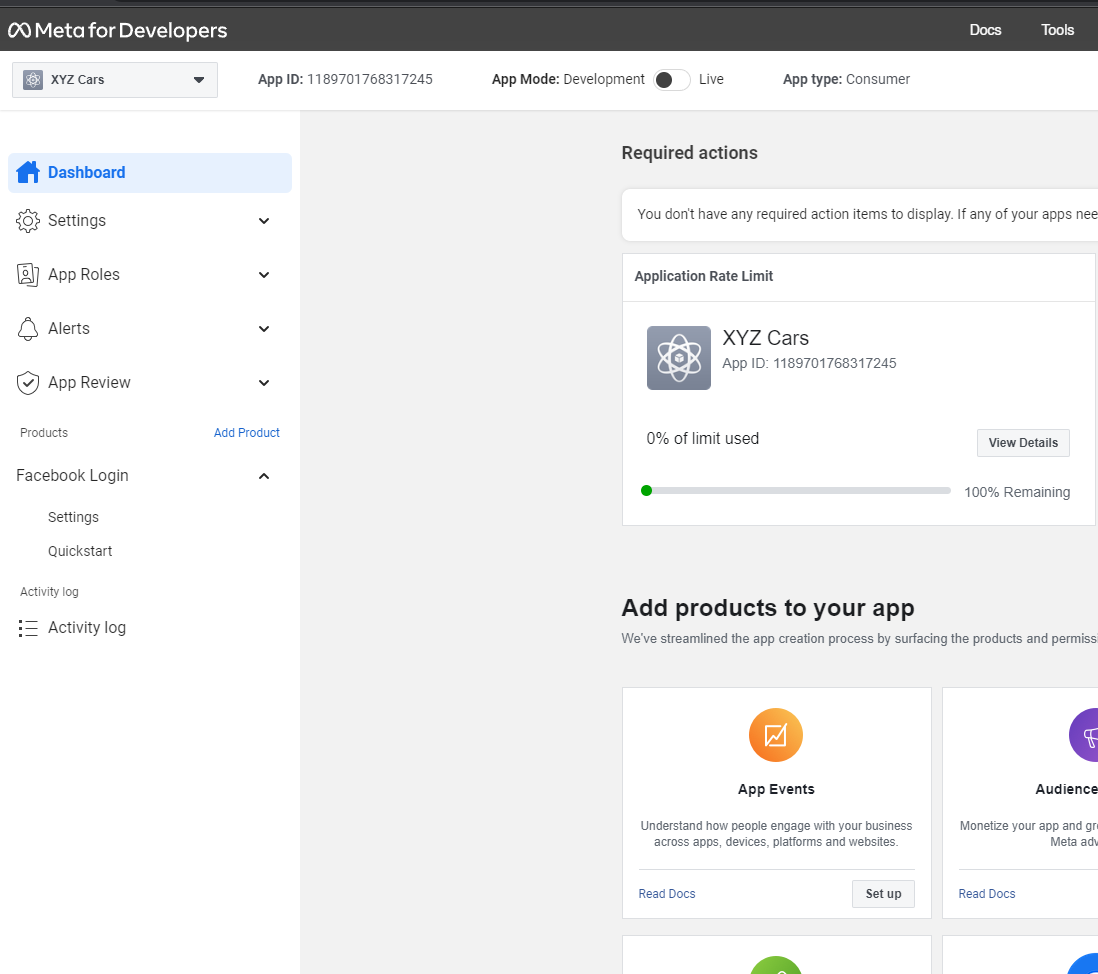
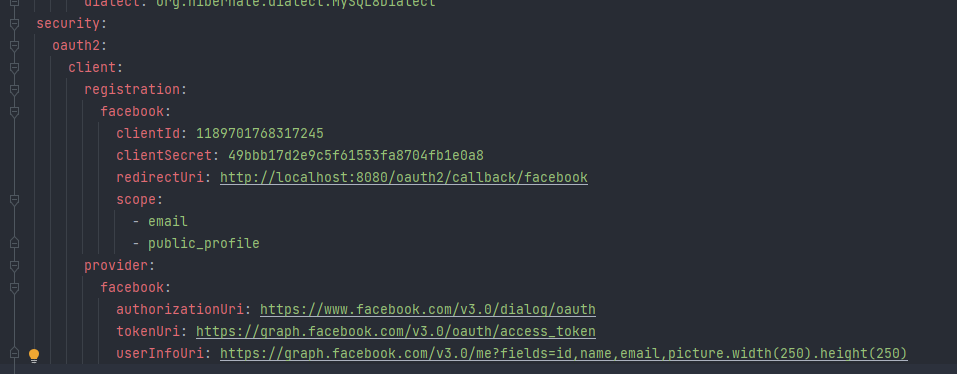
****

FacebookLogin.js  


LoginPage.js  




1. **Construct the application which implements the selected API in Task 2.**
2. **Using Facebook Login API**

* Create new developer application on facebook developer website  
  
* Add Product for Facebook Login   
  
* Use created App Id and Secret in Application  
  

1. Task 4

**Create the following items under “Application Testing” in Project Report**

1. Implement white Box testing for the developed API of your Application
2. Conduct Black Box testing (UAT testing) of your developed application and show the evidence for each test case.
3. Once the testing done check failed test cases and the reason to fail the same and implement your application accordingly.

**Solution:**

1. **Implement white Box testing for the developed API of you Application**

* What is white Box testing?

White box testing, also referred to as structural testing or code-based testing, is a testing technique that looks at the software's internal implementation and structure. It is a technique for evaluating a system's internal logic, data processing capabilities, and control flow. White box testing's primary goal is to make sure that the code adheres to design specifications, is free of bugs, and has perfect syntax.

* Why we using white box testing?

White box testing is used to:

• Ensure that each independent path within a module is executed at least once.

• Discover logic or design errors that may occur during the actual implementation.

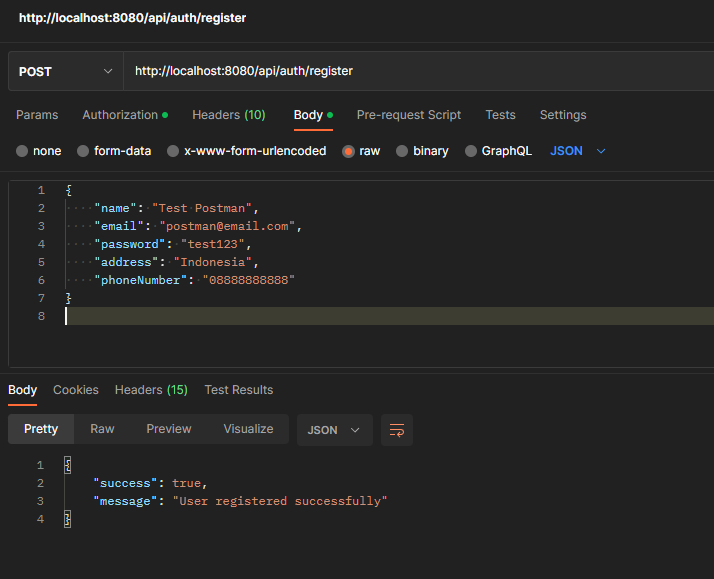
• Identify security holes or vulnerabilities in your code.

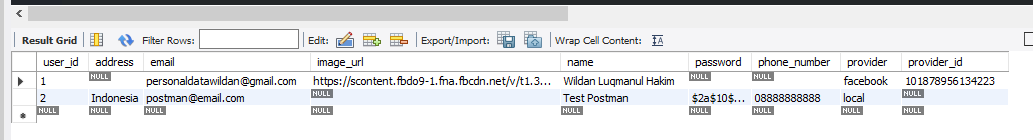
• Check for typos and grammatical errors.

Testing APIs with Postman

Register

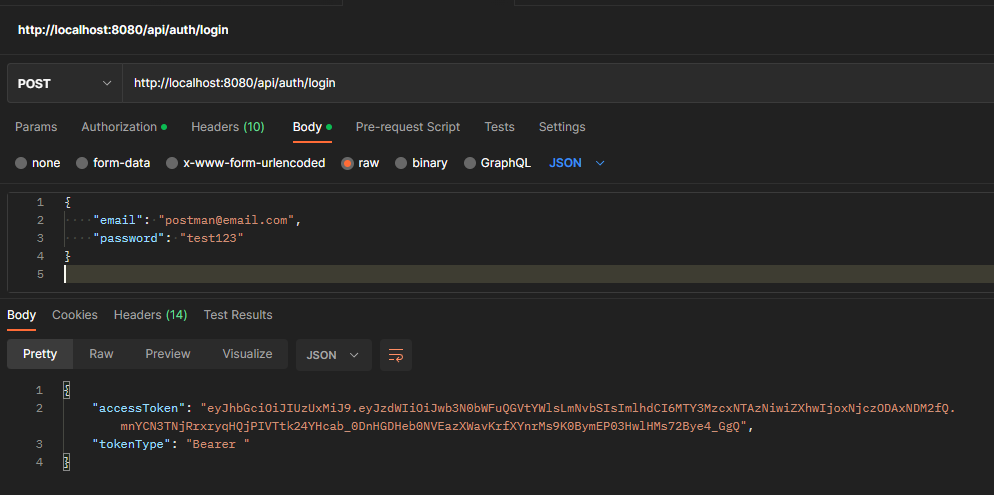
<http://localhost:8080/api/auth/register>





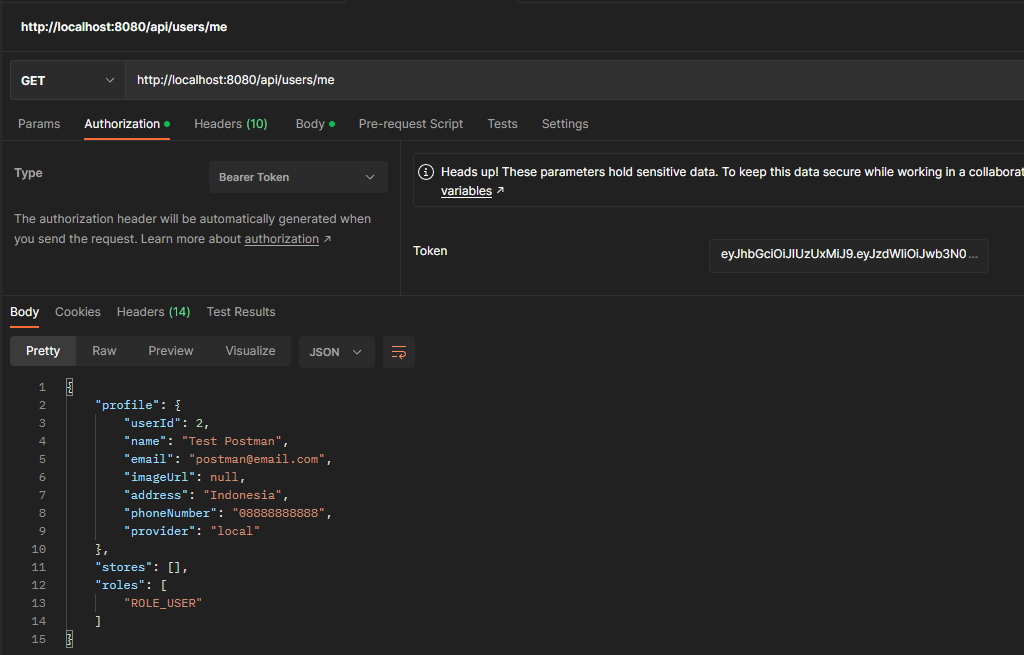
Login

<http://localhost:8080/api/auth/login>



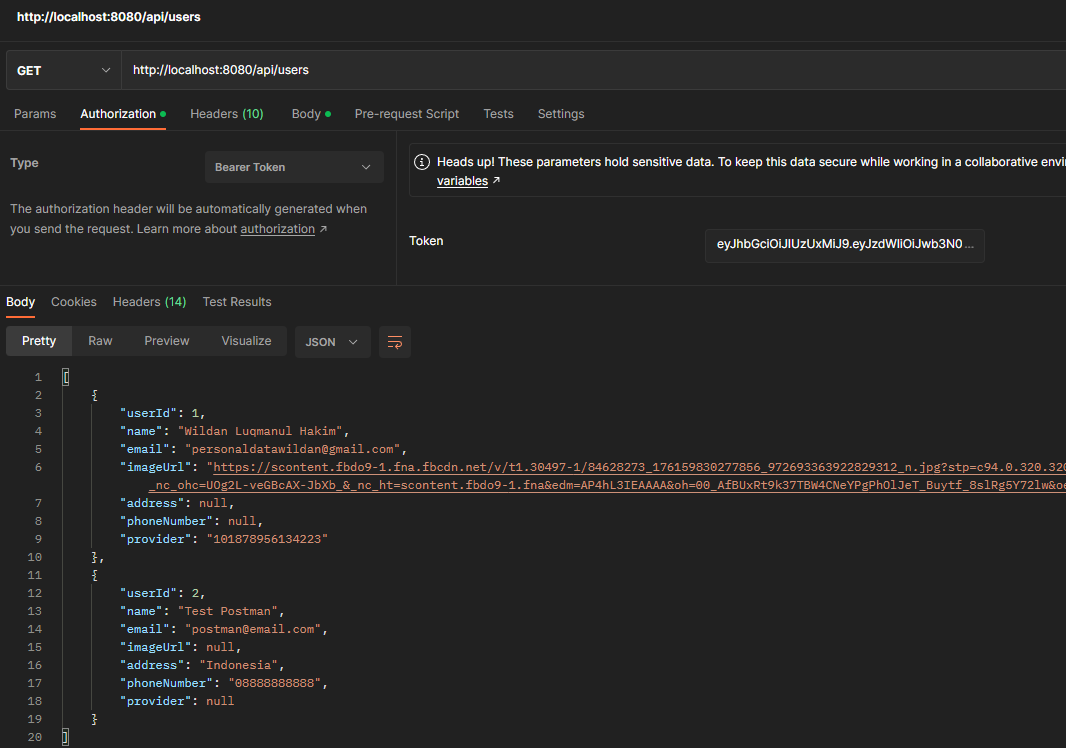
Get User Login

<http://localhost:8080/api/users/me>



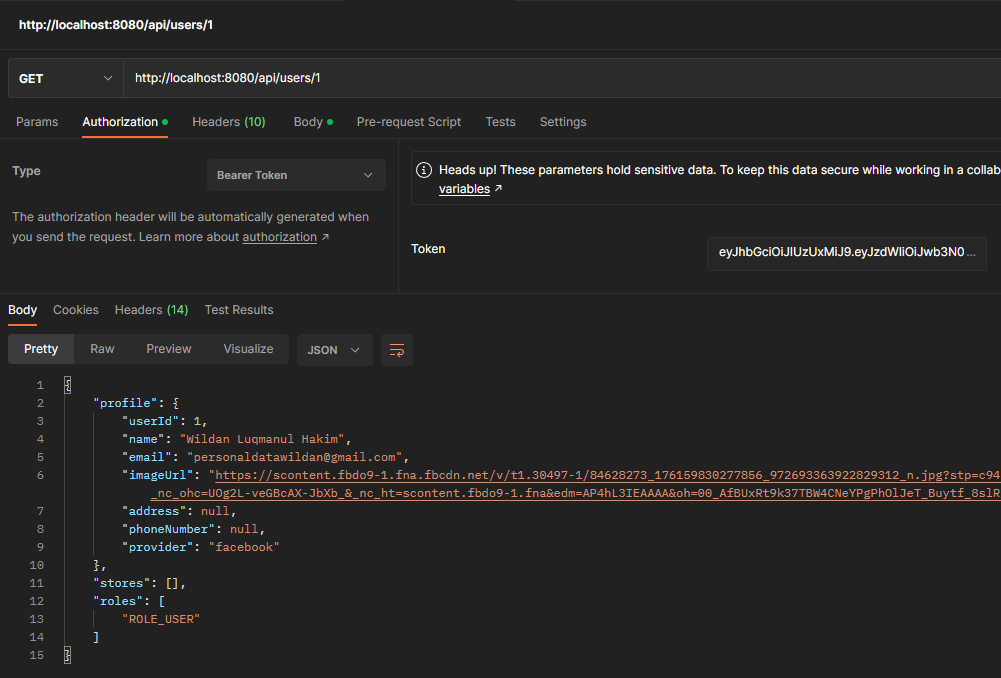
Get List User

<http://localhost:8080/api/users>



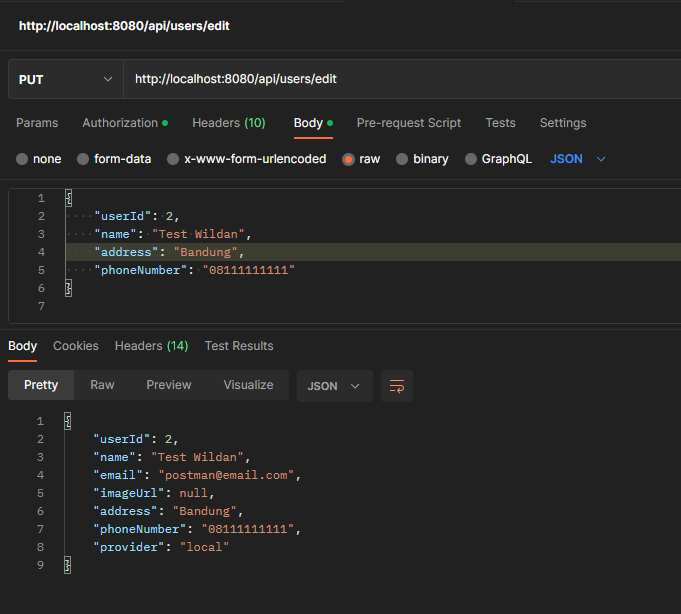
Get User by Id

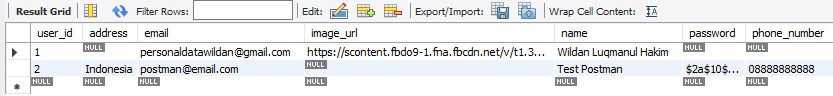
[http://localhost:8080/api/users/{userId}](http://localhost:8080/api/users/%7buserId%7d)

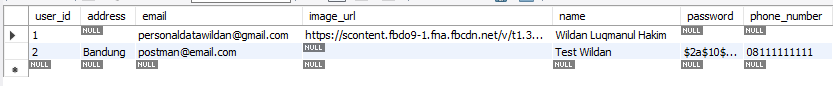


Edit Profile

<http://localhost:8080/api/users/edit>

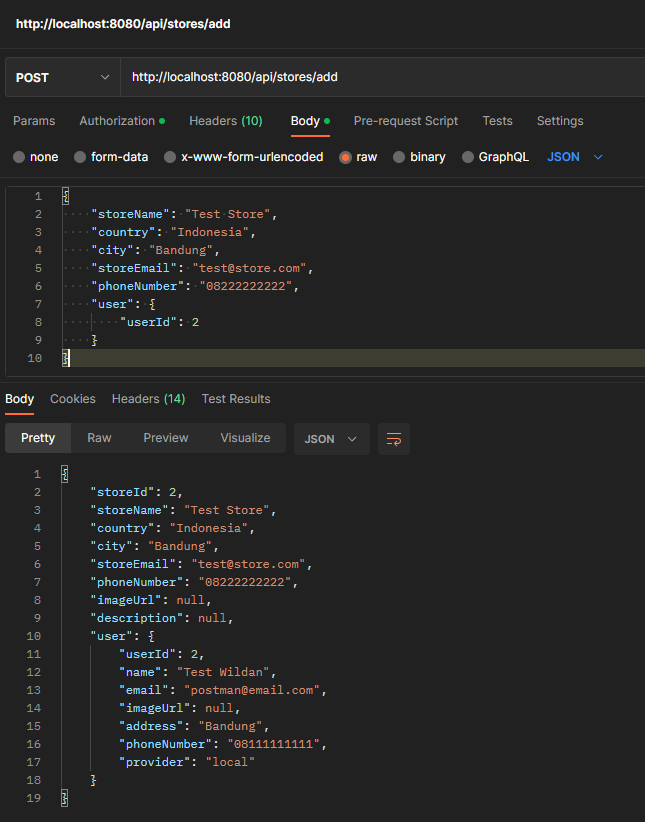


Before  


After  


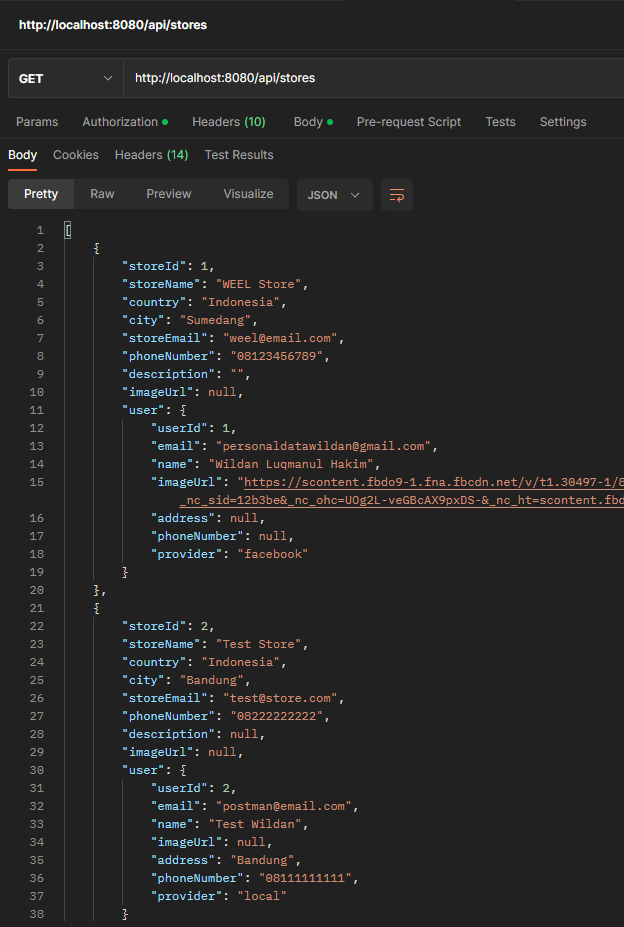
Add Store (need ROLE\_ADD\_STORE)

<http://localhost:8080/api/stores/add>



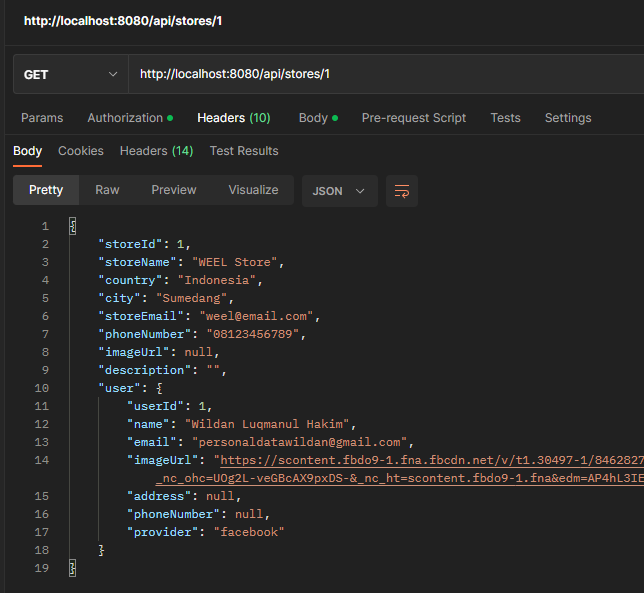
Get List Store

<http://localhost:8080/api/stores>



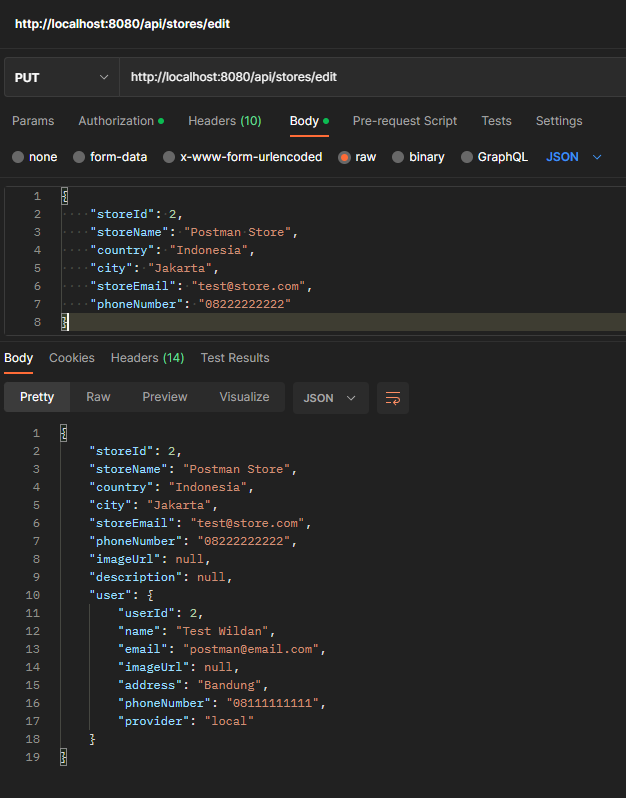
Get Store by Id

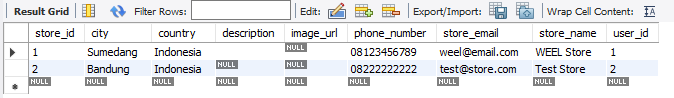
[http://localhost:8080/api/stores/{storeId}](http://localhost:8080/api/stores/%7bstoreId%7d)

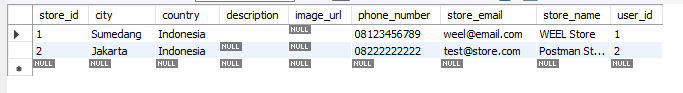


Edit Store (need ROLE\_ADD\_STORE)

<http://localhost:8080/api/stores/edit>



Before  


After  


1. **Conduct Black Box testing (UAT testing) of your developed application and show the evidence for each test case.**

|  |  |
| --- | --- |
| **Test Scenario** | Registration |
| TS001 |
| **Test Cases** | Registration in KYN Website |
| TC001 | User input forms and not fill one forms and it should show danger alert |
| TC002 | User input data into forms and it should show success alert |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Priority** | **Pre-conditions** | **Expected Result** | **Actual Result** | **Final Result** |
| TS001 | TC001 | High | - User visit KYN Website  -User open register page | User not input password, it should show an error alert | As expected | Pass |
| TC002 | High | User input all forms, it should show a success alert | As expected | Pass |

**Test Data Table**

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Data** |
| TS001 | TC001 | Email = test@email.com  Password = (not inputed)  Name = Wildan  Address = Sumedang  Phone number = 088888888 |
| TC002 | Email = test@email.com  Password = test123  Name = Wildan  Address = Sumedang  Phone number = 088888888 |

**Test Evidences=**

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Evidences** |
| TS001 | TC001 |  |
| TC002 |  |

|  |  |
| --- | --- |
| **Test Scenario** | Login |
| TS001 |
| **Test Cases** | Login in KYN Website |
| TC001 | Login with account that have registered |
| TC002 | Login with facebook |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Priority** | **Pre-conditions** | **Expected Result** | **Actual Result** | **Final Result** |
| TS001 | TC001 | High | - User visit KYN Website  -User open login page | User input email and password to login into website, user should be redirect to profile page | As expected | Pass |
| TC002 | High | User click login with facebook button into website, user should be redirect to profile page | As expected | Pass |

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Data** |
| TS001 | TC001 | Email = [test@email.com](mailto:test@email.com)  Password = test123 |
| TC002 | “Login with Facebook account” |

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Evidences** |
| TS001 | TC001 |  |
| TC002 |  |

1. Task 5

**Create the following items under “Review and Reflect on the APIs Used” in Project Report**

1. Review your developed API, identify the strength and weaknesses of API.
2. Provide data security report of your developed application.

**Solution:**

1. **Review your developed API, identify the strength and weaknesses of API.**

**Strengths and Weakness selected API, features**

Strength:

* A wide range of developers can use them because they are easy to use and comprehend.
* Since they transmit data using HTTP standard methods (such as GET, POST, PUT, and DELETE), they are simple to integrate with other tools and systems.
* Because of their light weight, they can be used with a variety of networks and devices.
* They don't require the maintenance of sessions or connections because they are stateless, which means that each request can stand on its own.
* Because of their platform independence, they work with a wide range of frameworks and programming languages.

Weakness:

* Because they rely on the underlying transport protocol (HTTP) for security, they may be less secure than other kinds of APIs.
* Because they depend on users' accurate input, they may be vulnerable to mistakes if not implemented properly.
* Because they need to parse JSON or XML to process data, they may be less effective than other types of APIs.
* Some use cases might call for more intricate features than REST can offer.

1. **Provide data security report of your developed application.**

The backend KYN app is secured by Spring Security. Spring Security offers smart and scalable security features such as authentication and authorization.

**Spring Security**

Java-based application security is handled by the Spring Security framework. For both web and non-web applications, it offers a complete security solution that includes support for authentication, authorization, and access-control.

Spring Security has a number of important features, including:

* Username/password, certificate-based, and token-based authentication are just a few of the many authentication methods that Spring Security supports.
* Authorization: Using method-level and expression-based access control, Spring Security enables the creation of adaptable and granular access control for both web and non-web resources.

**Authentication with JWT**

JSON Web Token (JWT) is a technique for user authentication that involves supplying a signed token with a JSON payload containing user data. Each request includes a token that is sent to the server, and the server can check the signature of the token to ensure its validity. JWTs are frequently used in stateless authentication systems, in which client sessions are entirely stored on the client side in JWT files rather than on the server.

This project decided to use JWT for authentication for a number of reasons:

* JWT are self-sufficient: A JWT is a small, efficient token that contains all the data required to authenticate a user. As a result, there is no need to keep authentication data in a session or on the server, which can make the application's implementation easier.
* JWT are safe: Since JWTs are signed with a secret key, it is challenging for attackers to forge them or alter their contents.
* JWT have a lot of support: It is simple to incorporate JWT into your application because it is a standard that is supported by many different platforms and libraries.

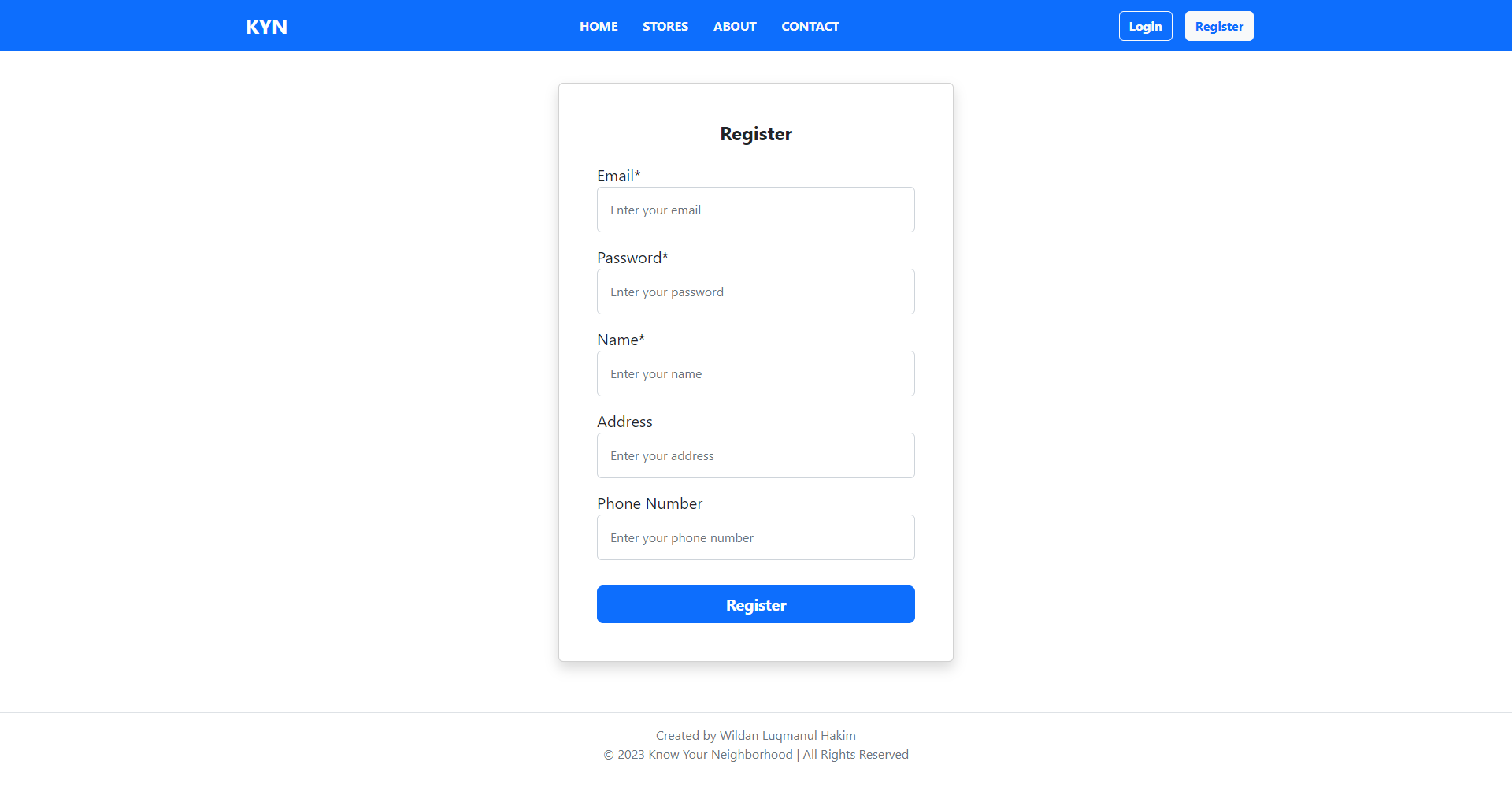
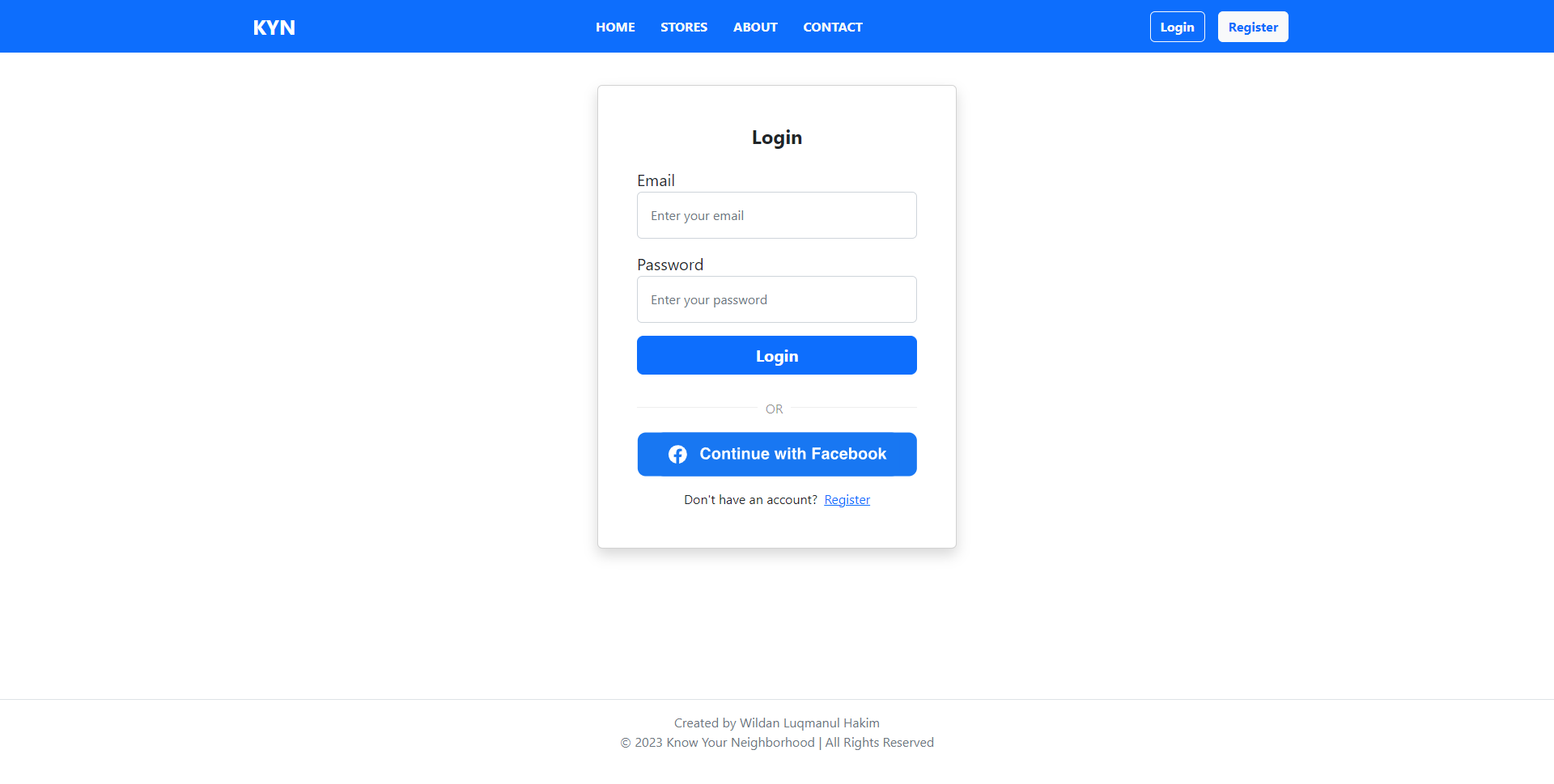
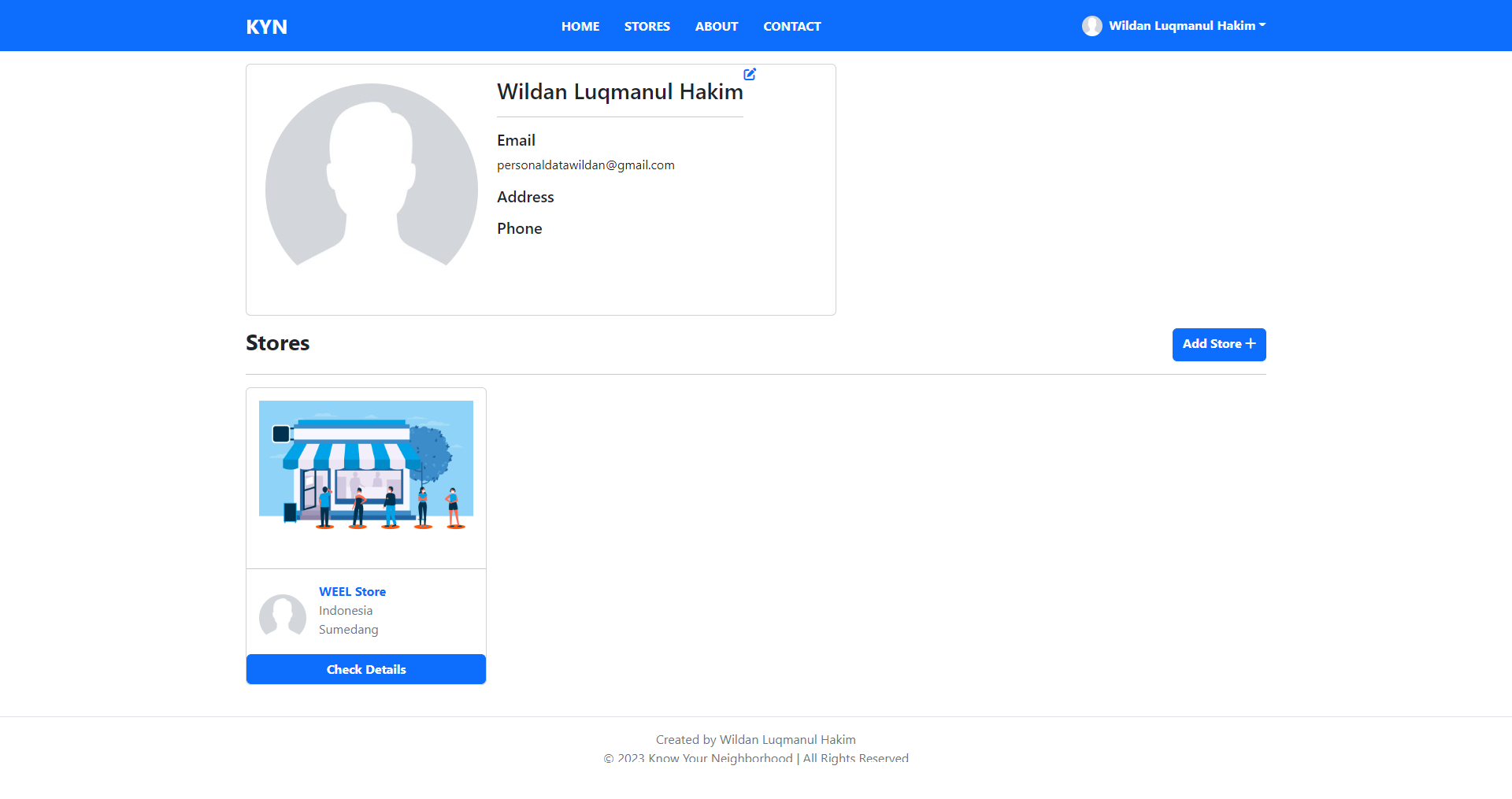
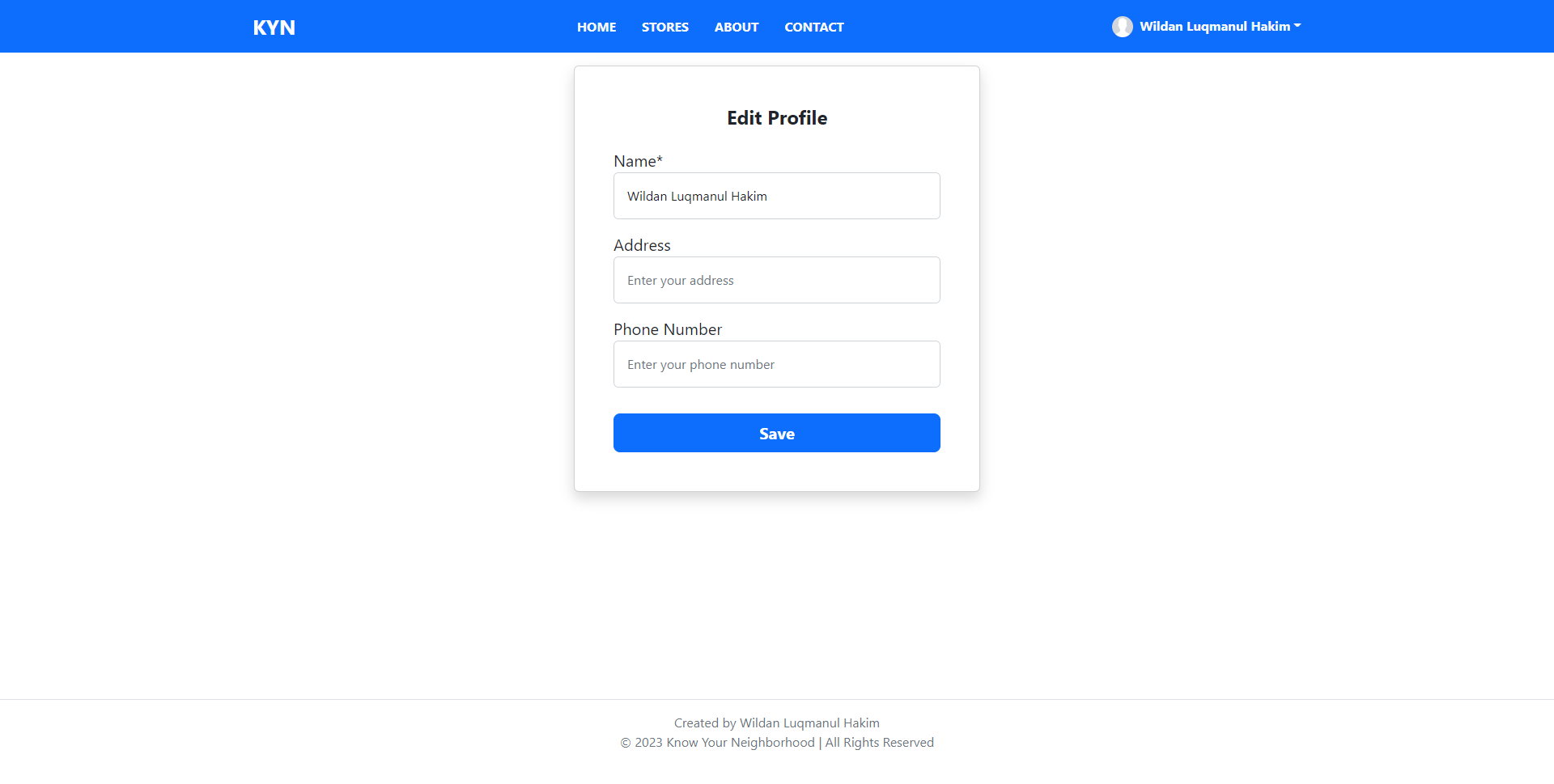
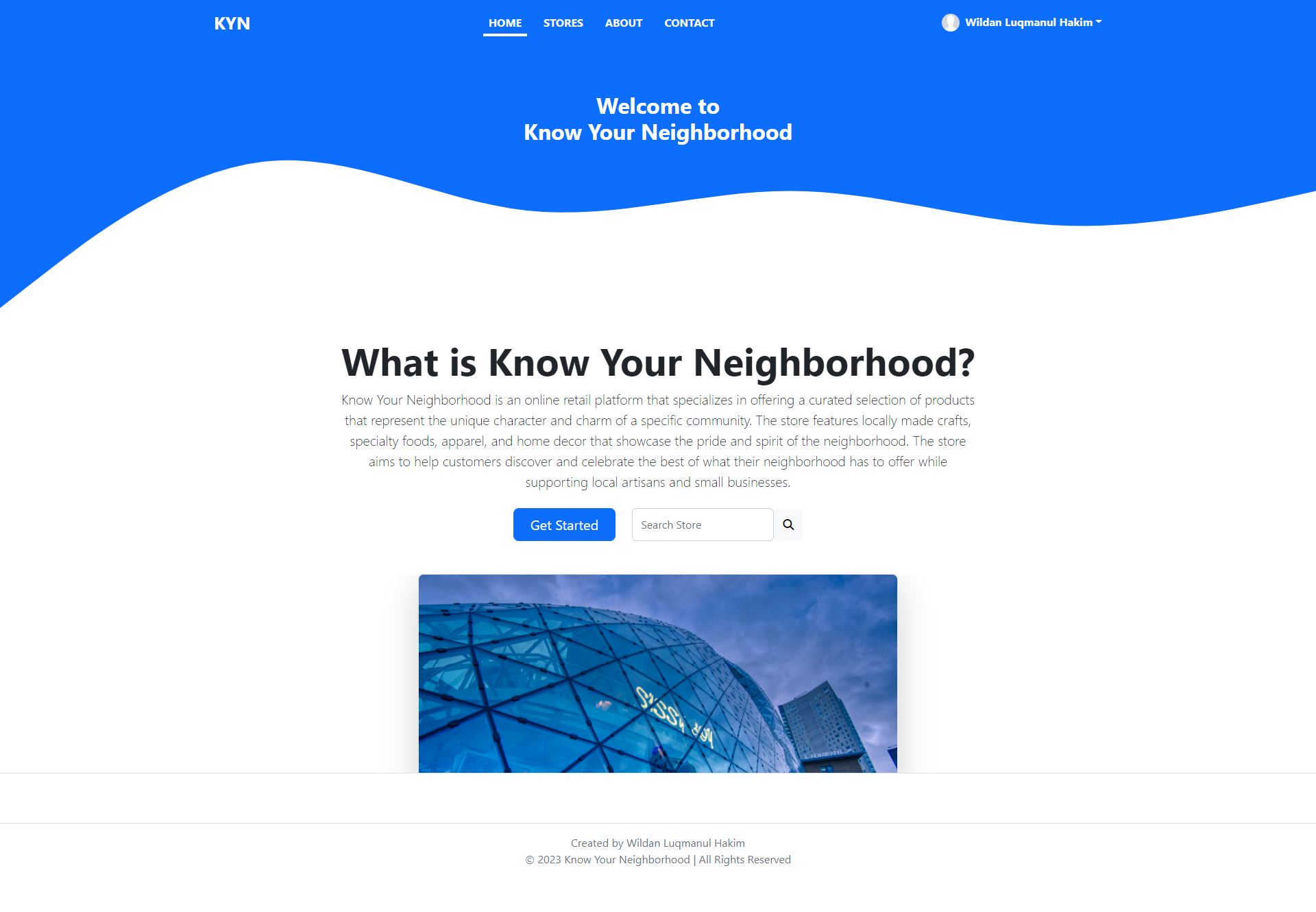
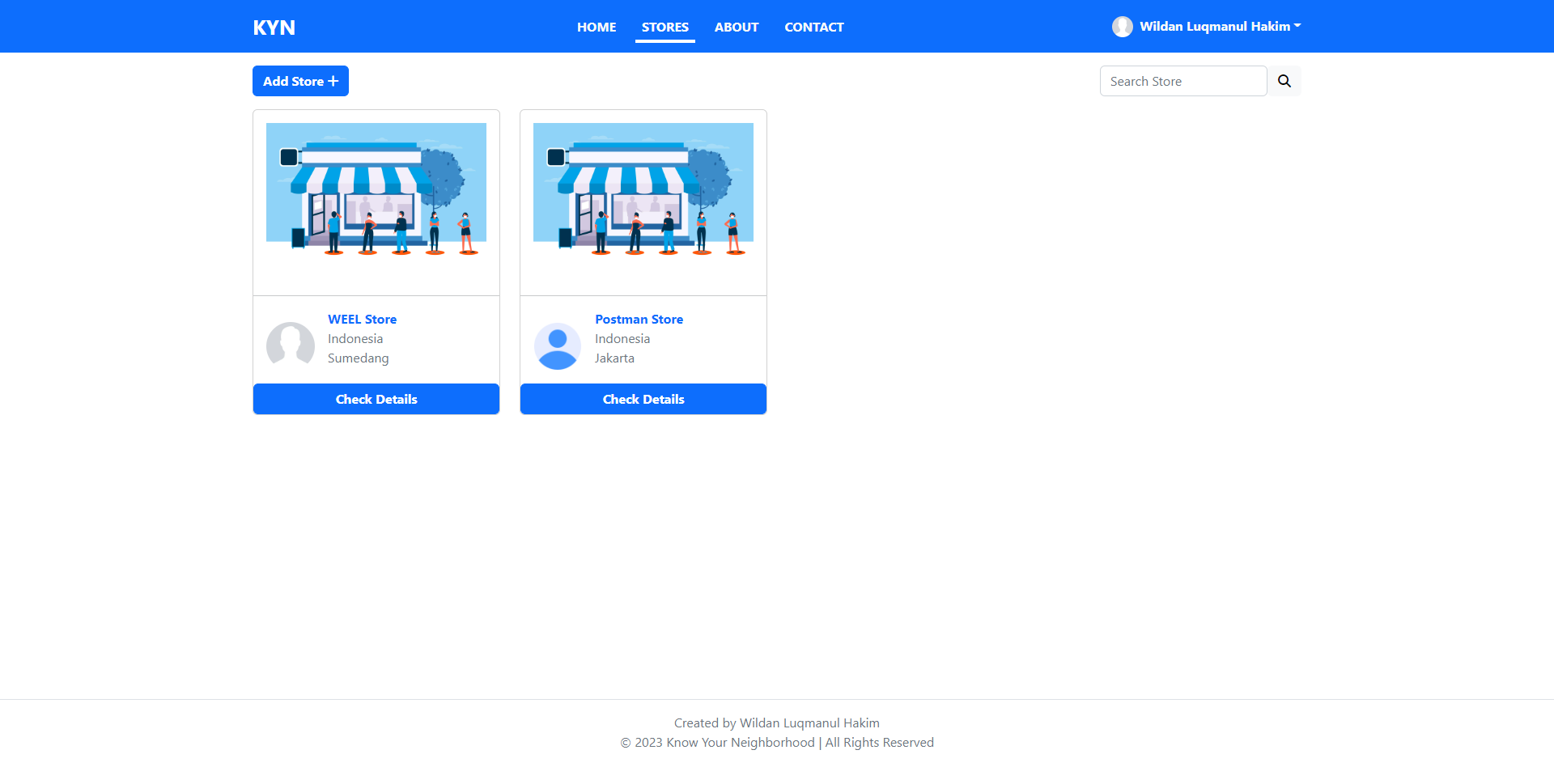
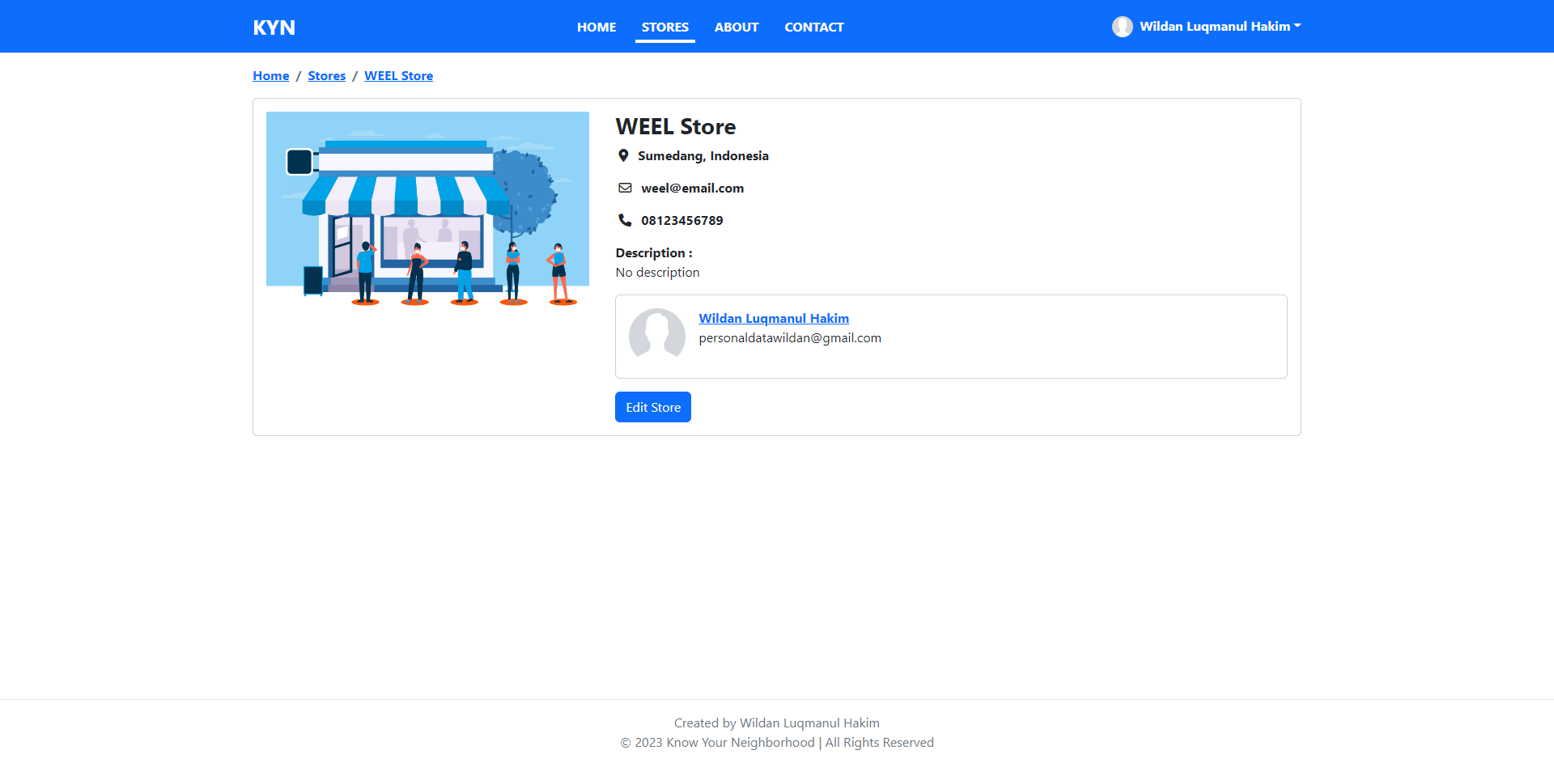
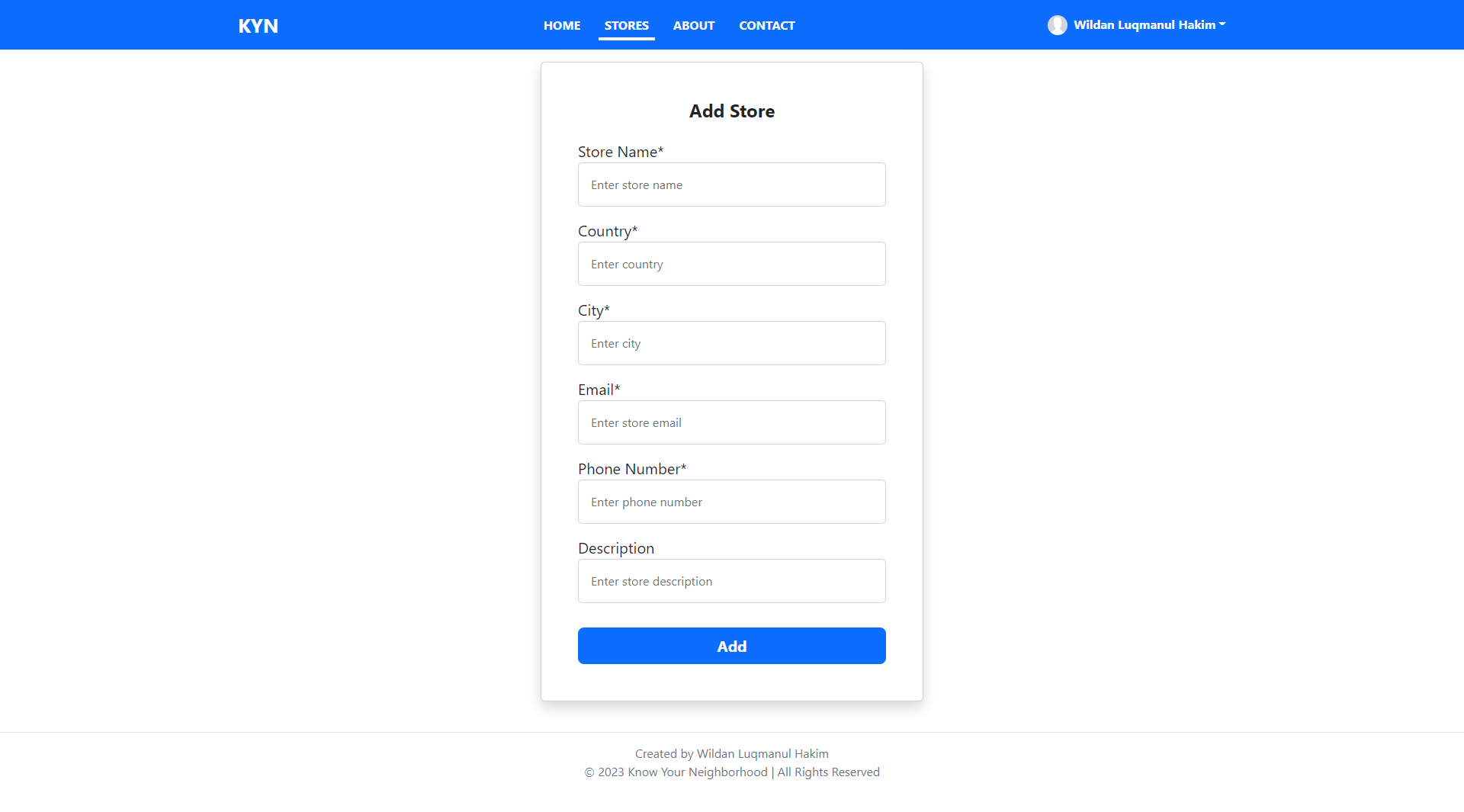
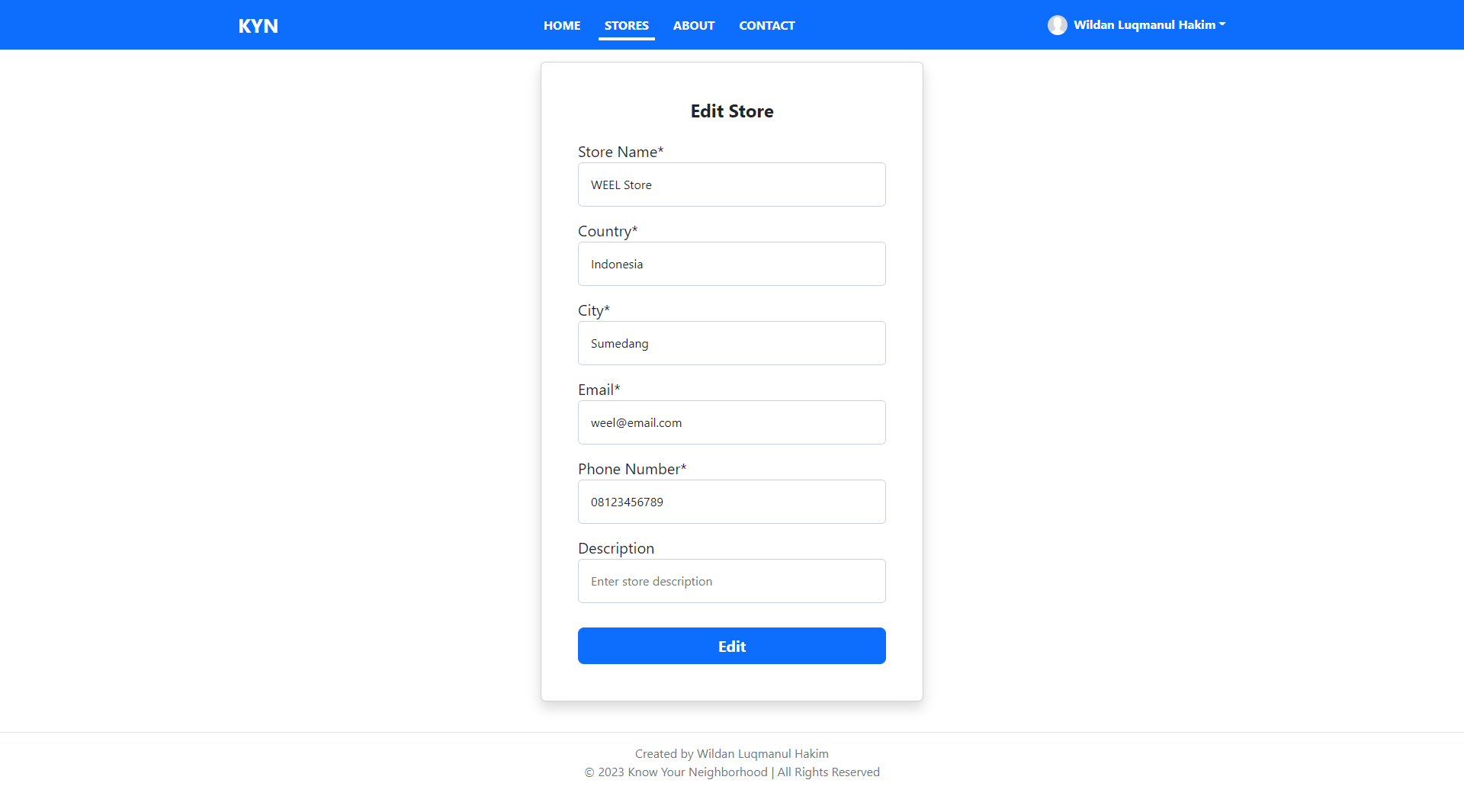
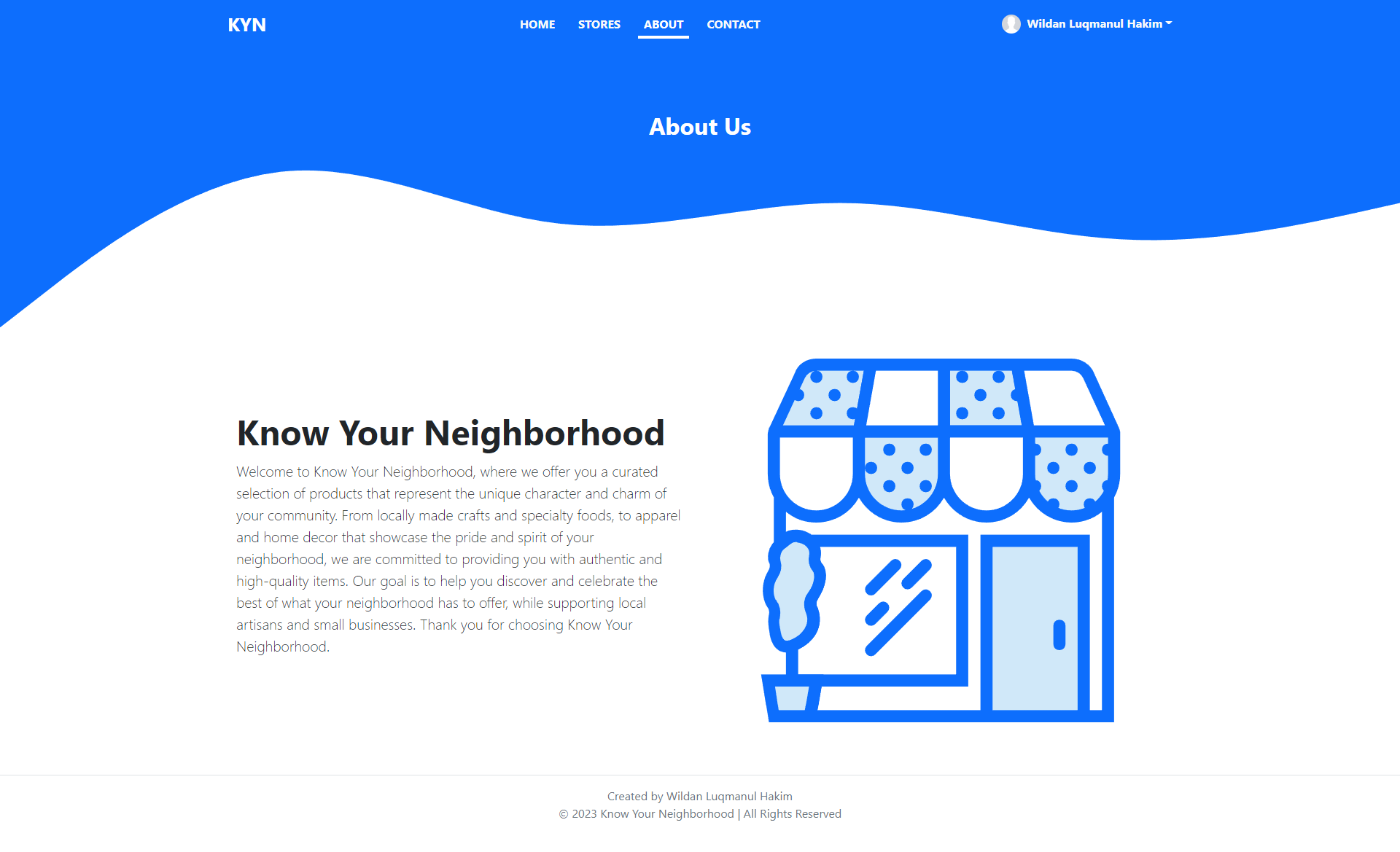
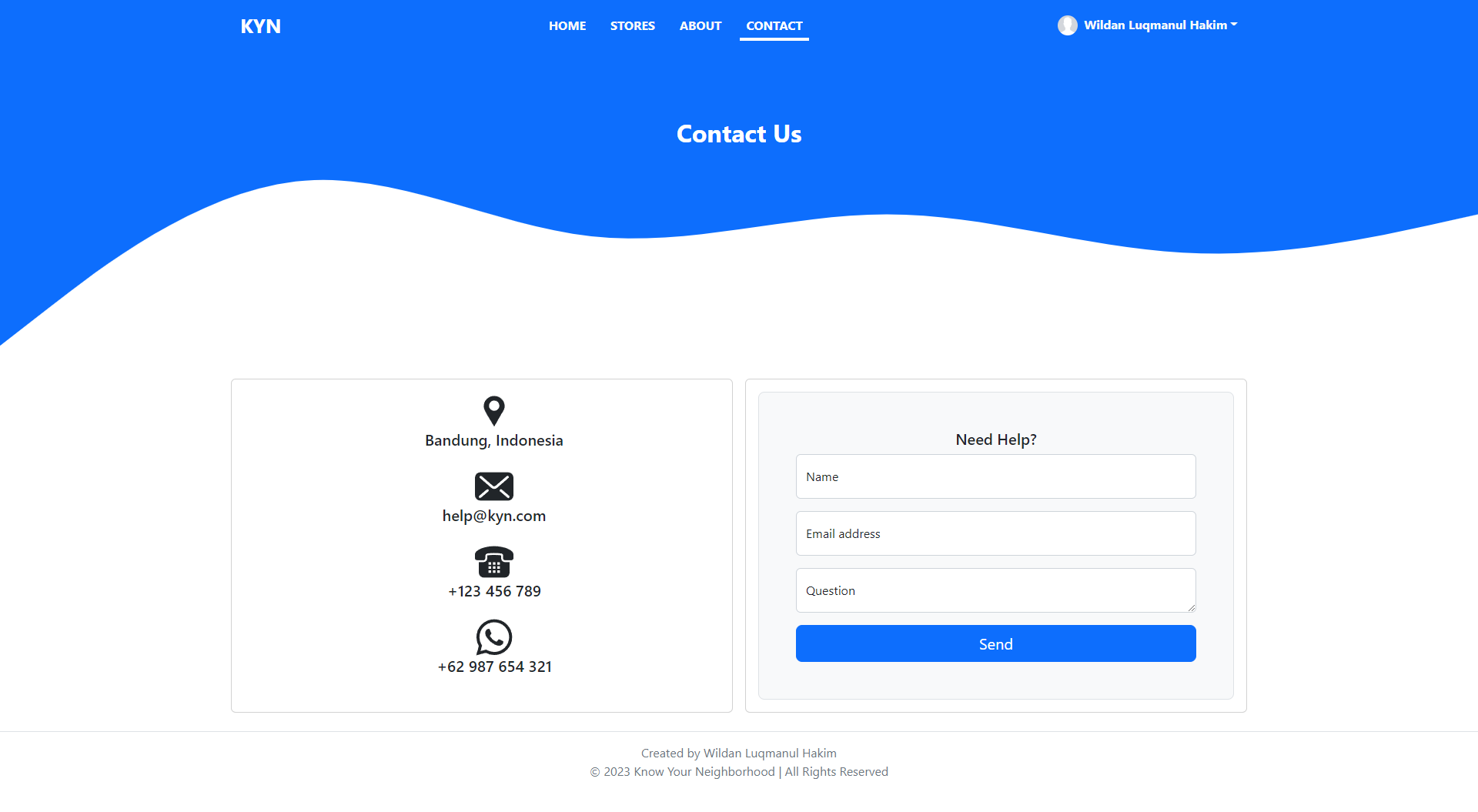
**Security with OAuth2**OAuth 2.0 (Open Authorization) is a free standard for authorization that enables users to share their confidential resources (such as data and files) stored on one website with another without disclosing their login information (username and password).

1. Task 6

Task Statement:

1. Provide screen capture of developed application using APIs in Project Presentation

Solution:

1. **Register Page  
   **
2. **Login Page  
   **
3. **Profile Page  
   **
4. **Edit Profile Page  
   **
5. **Home Page  
   **
6. **Stores Page  
   **
7. **Detail Store Page  
   **
8. **Add Store Page  
   **
9. **Edit Store Page  
   **
10. **About Us Page  
    \**
11. **Contact Us Page  
    **