Project Report

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

|  |  |  |  |
| --- | --- | --- | --- |
| Student name | | Assessor name | |
| Ida Bagus Ketut Yoghantara | |  | |
| Date issued | Completion date | | Submitted on |
| 2 January 2023 | 11 January 2023 | | 11 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: 2 January 2023 |

Content

1. Project background
2. Project Objectives
3. Project Requirement Specifications
4. Task 1
5. Task 2
6. Task 3
7. Task 4
8. Task 5
9. Task 6

Project Background

The Know-Your-Neighborhood application was developed using Spring Boot. The goal of this project is to rebuild the application using React JS and Spring Boot and use existing APIs to log in and out. Your application should have a login button with an API available. Also, the scope of the project is to examine and compare different existing APIs sample by sample, assess their suitability, identify potential security issues, and use selected APIs on existing websites to create a login.

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

Project Requirements Specifications

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

Task 1

Task Statement:

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.

What is API?

An API, or Application Programming Interface, is a set of rules that specifies how two software programs should interact with each other. It serves as an intermediary between different software systems, allowing them to communicate with each other and share data and functionality. APIs allow developers to build software applications that can make use of certain features or data of another application, without having to know how the other application is implemented.

API role and need for API

APIs are used in a wide variety of contexts, including operating systems, application software, and web-based systems. They can be used to allow different software programs to communicate with each other, to allow a front-end web application to access data from a back-end database, or to enable a mobile app to connect to a network-based service.

The need for APIs arises because different software systems often need to be integrated with each other in order to exchange data or provide certain services. For example, a mobile app that helps users find nearby restaurants might use an API to fetch data from a database of restaurant listings. Without an API, the mobile app would have to directly access the database, which would be difficult to implement and maintain.

Research existing APIs

There are many different APIs available for a wide range of purposes. Some APIs are open and available to anyone, while others are proprietary and only available to specific organizations or individuals. To research existing APIs, you can search online directories such as ProgrammableWeb or RapidAPI, which list thousands of APIs in various categories. You can also search for APIs specific to a particular service or technology, such as the Google Maps API or the Facebook API.

1. Examine the relationship between API and SDK.

**API**, or Application Programming Interface, is a set of rules that specifies how two software programs should interact with each other. It serves as an intermediary between different software systems, allowing them to communicate with each other and share data and functionality.

**SDK**, or Software Development Kit, is a collection of tools that developers can use to build software applications. It typically includes libraries, documentation, and sample code that developers can use as a starting point for their own projects.

**The** **relationship** between **API** and **SDK** is that SDK typically includes an API as part of its tools. The API is a set of programming instructions that specifies how the software should interact with other systems, while the SDK provides a more comprehensive set of tools that developers can use to build software applications that use the API. In this way, the API is a component of the SDK, and the SDK provides a more complete set of tools for working with the API.

1. Identify types of API and its uses.

There are several types of APIs, each with its own specific use cases:

1. **Open APIs**, also known as External or External Partner APIs, are available to developers outside the organization that provides the API. They are typically used to allow developers to access certain data or functionality of an application, such as the Google Maps API, which allows developers to include maps and location-based functionality in their own applications.
2. **Internal APIs**, also known as Private or Partner APIs, are only available to developers within the same organization that provides the API. They are used to allow different teams within an organization to communicate with each other and share functionality and data.
3. **Composite APIs** allow developers to access multiple endpoints in a single API call, making it easier to request data from multiple sources.
4. **REST APIs** use the HTTP protocol to send data between different systems. They are lightweight and easy to use, making them a popular choice for building APIs.
5. **SOAP APIs** use the Simple Object Access Protocol to exchange data between systems. They are more powerful and flexible than REST APIs, but also more complex to use.

APIs are used in a wide range of applications, such as mobile apps, web applications, and Internet of Things (IoT) devices. They allow developers to build software that can communicate with other systems, access data, and provide certain services to users.

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

There are several potential security issues that can arise with APIs:

1. Injection attacks: APIs are vulnerable to injection attacks, such as SQL injection or code injection, if they do not properly validate or sanitize input data.
2. Authentication and authorization: APIs should have proper authentication and authorization measures in place to ensure that only authorized users can access the API and its data.
3. Sensitive data exposure: APIs may expose sensitive data, such as passwords or personal information, if they do not adequately protect this data.
4. Lack of rate limiting: APIs that do not have rate limiting in place may be vulnerable to denial-of-service attacks or other types of abuse.
5. Lack of transport encryption: APIs should use secure transport protocols, such as HTTPS, to protect data in transit.

To evaluate the suitable API for a given scenario or application, we should consider the following factors:

1. Security: The API should have robust security measures in place to protect against potential vulnerabilities.
2. Compatibility: The API should be compatible with the technology stack of the application you are building.
3. Functionality: The API should provide the necessary functionality for your application.
4. Performance: The API should have good performance, with low latency and high availability.
5. Documentation: The API should have comprehensive documentation that is easy to understand and use.
6. Support: The API should have good support from the provider, with resources such as tutorials, forums, and email support.
7. Pricing: The API should have a pricing model that is suitable for your needs and budget.

Task 2

Task Statement:

1. Analyze the given scenario, identify the requirements, and select the suitable API for the same.
2. Develop the relevant wireframes to utilize the API for given purpose.
3. Identify the scope and target platforms.
4. Evaluate and justify the selection of chosen APIs for the application. (Show security of selected APIs.)

Solution:

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

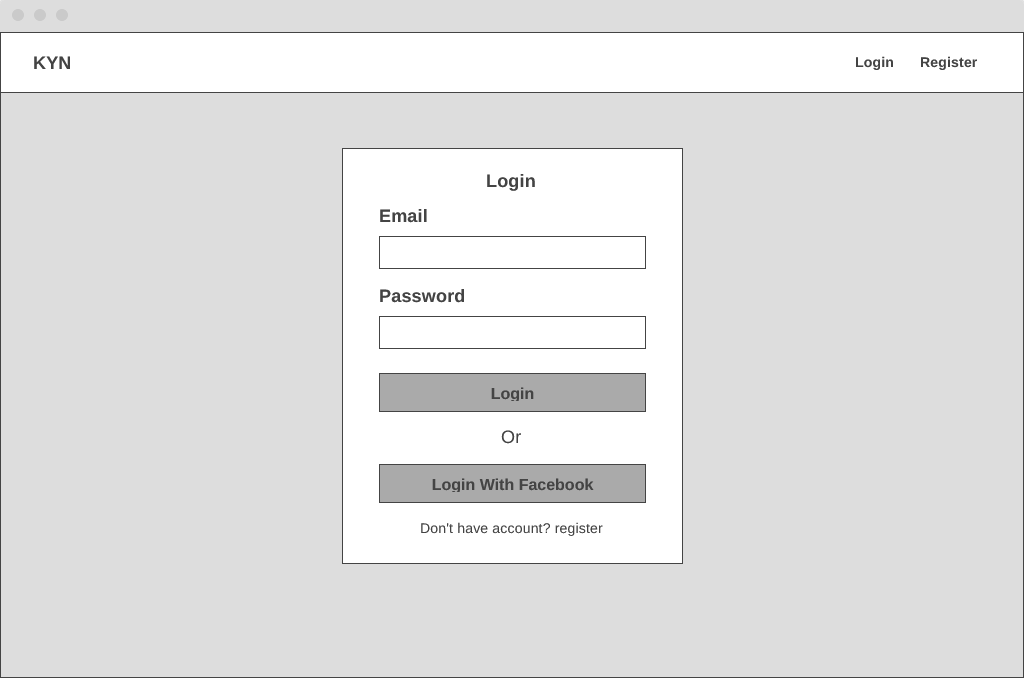
Facebook API

Facebook Login is a fast and convenient way for people to create accounts and log into your app across multiple platforms. It's available on [iOS](https://developers.facebook.com/docs/ios/login/), [Android](https://developers.facebook.com/docs/android/login-with-facebook), [Web](https://developers.facebook.com/docs/facebook-login/web), [desktop apps](https://developers.facebook.com/docs/facebook-login/manually-build-a-login-flow) and [devices such as Smart TVs and Internet of Things objects](https://developers.facebook.com/docs/facebook-login/for-devices).

Facebook Login enables two scenarios, [authentication](https://developers.facebook.com/docs/facebook-login/auth-vs-data/#authentication) and asking for [permissions](https://developers.facebook.com/docs/facebook-login/permissions/overview) to [access people's data](https://developers.facebook.com/docs/facebook-login/auth-vs-data/#data-access). You can use Facebook Login simply for authentication or for both authentication and data access. The user can login using their Facebook account, API will display and use their profile for interact with the Know Your Neighborhood Website

1. Develop the relevant wireframes to utilize the API for given purpose.

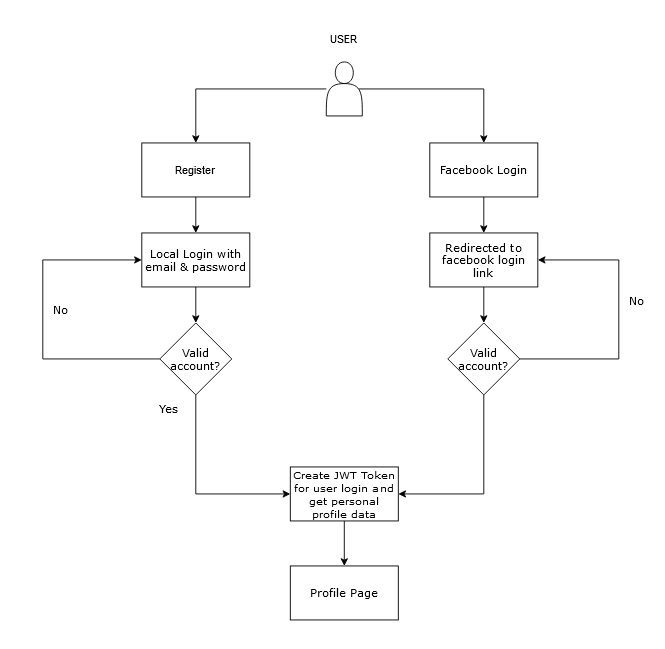
Facebook Login Button



1. Identify the scope and target platforms.

Facebook's APIs are now available on the majority of devices and operating systems. It can, for example, be used in both Windows and Mac environments, as well as on desktop and mobile devices.

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



User can choose to login with local login (provided by the website) or using facebook login to login into the website. If user choosing local login user can register first and login with email & password then redirected to profile page after successfully logged in. If user choosing facebook login, the user will be redirected to facebook login page and typing their credential, after user successfully login they will be redirected to their profile page.

Security of selected API for the application

OAuth is used as an open authorization protocol to grant restricted access to HTTP services to third-party applications acting on behalf of resource owners. It is capable of doing so without disclosing the user's identity or long-term credentials. It may also be used by a third-party application on its behalf. The OAuth working principal entails delegating user authentication to a service that hosts a user's account and granting access to the user's account to a third-party application.

Task 3

Task Statement:

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

Backend:

1. Spring Boot
2. Express JS
3. Django

Frontend:

1. React JS
2. Svelte
3. Vue JS

API implementation process:

1. REST API
2. SOAP API
3. OPEN API
4. Discuss a range of suitable development environments for front-end and back-end to develop an application

**Front-end development:**

* Visual Studio Code: A popular code editor that has a variety of extensions available to support front-end development, including support for HTML, CSS, and JavaScript.
* WebStorm: A dedicated front-end development environment with support for HTML, CSS, and JavaScript, as well as a built-in debugger and testing tools.

**Back-end development**

* PyCharm: A Python-specific development environment that includes support for web development frameworks like Django.
* Visual Studio: A general-purpose development environment that can be used for back-end development with a variety of languages, including C# and .NET.
* IntelliJ IDEA: A Java-specific development environment that can be used for back-end development with frameworks like Spring.

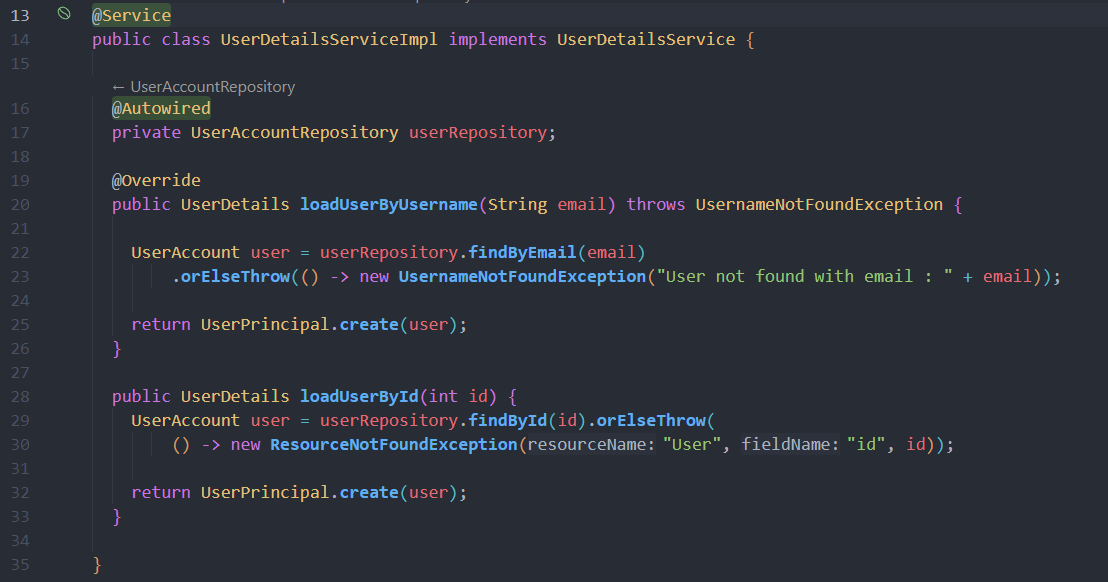
1. Develop a backend and Web service using selected development environment for given scenario

Backend Local Login

UserPrincipal.java



UserDetailsServiceImpl.java



AuthController.java





Backend Facebook Login

FacebookOAuth2UserInfo.java



HttpCookieOAuth2AuthorizationRequestRepository.java



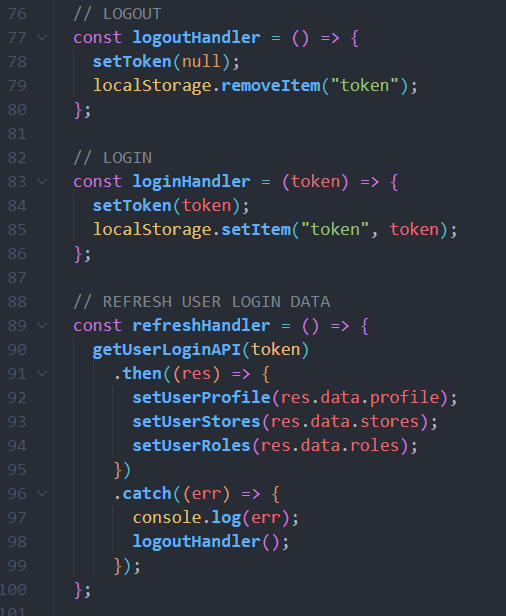
1. Develop an application that utilizes an API.

Developed Application using React

Auth-context.js







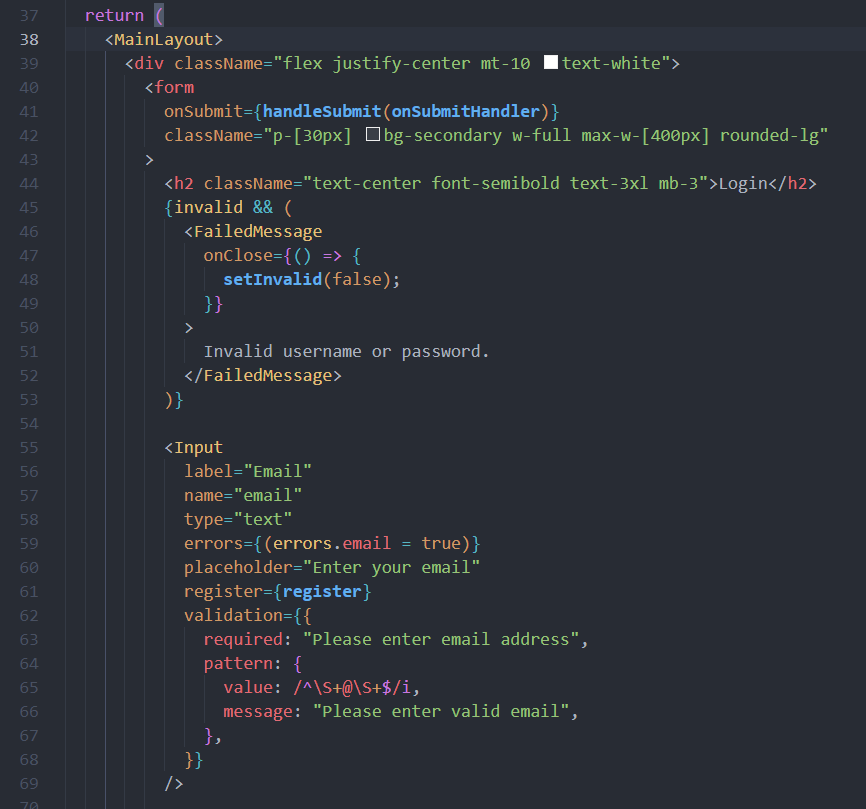


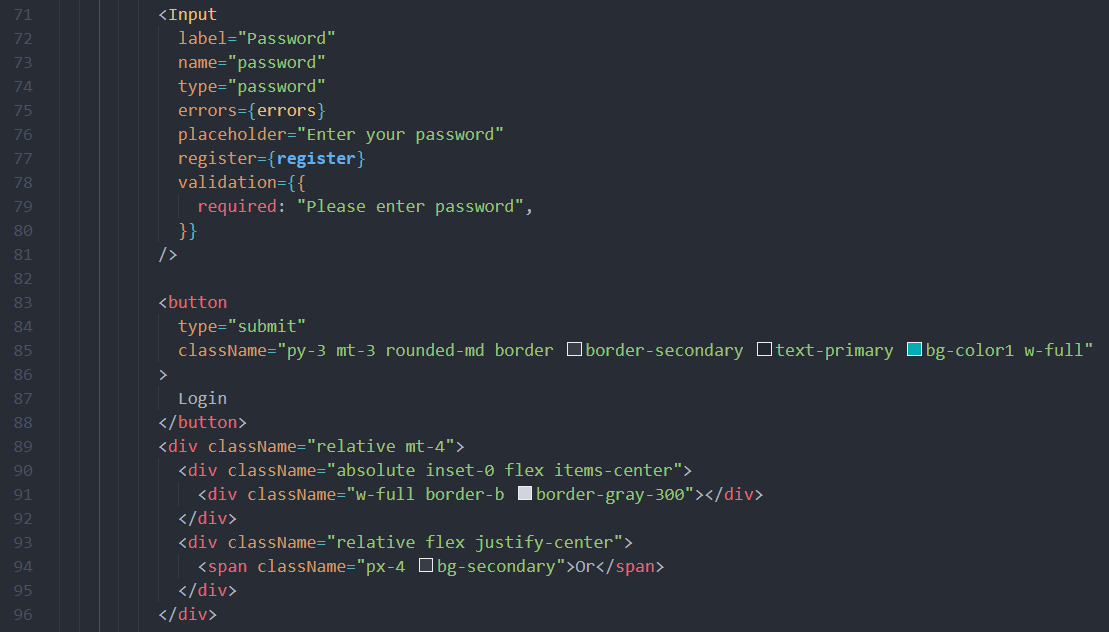
FacebookLogin.jsx



LoginPage.jsx





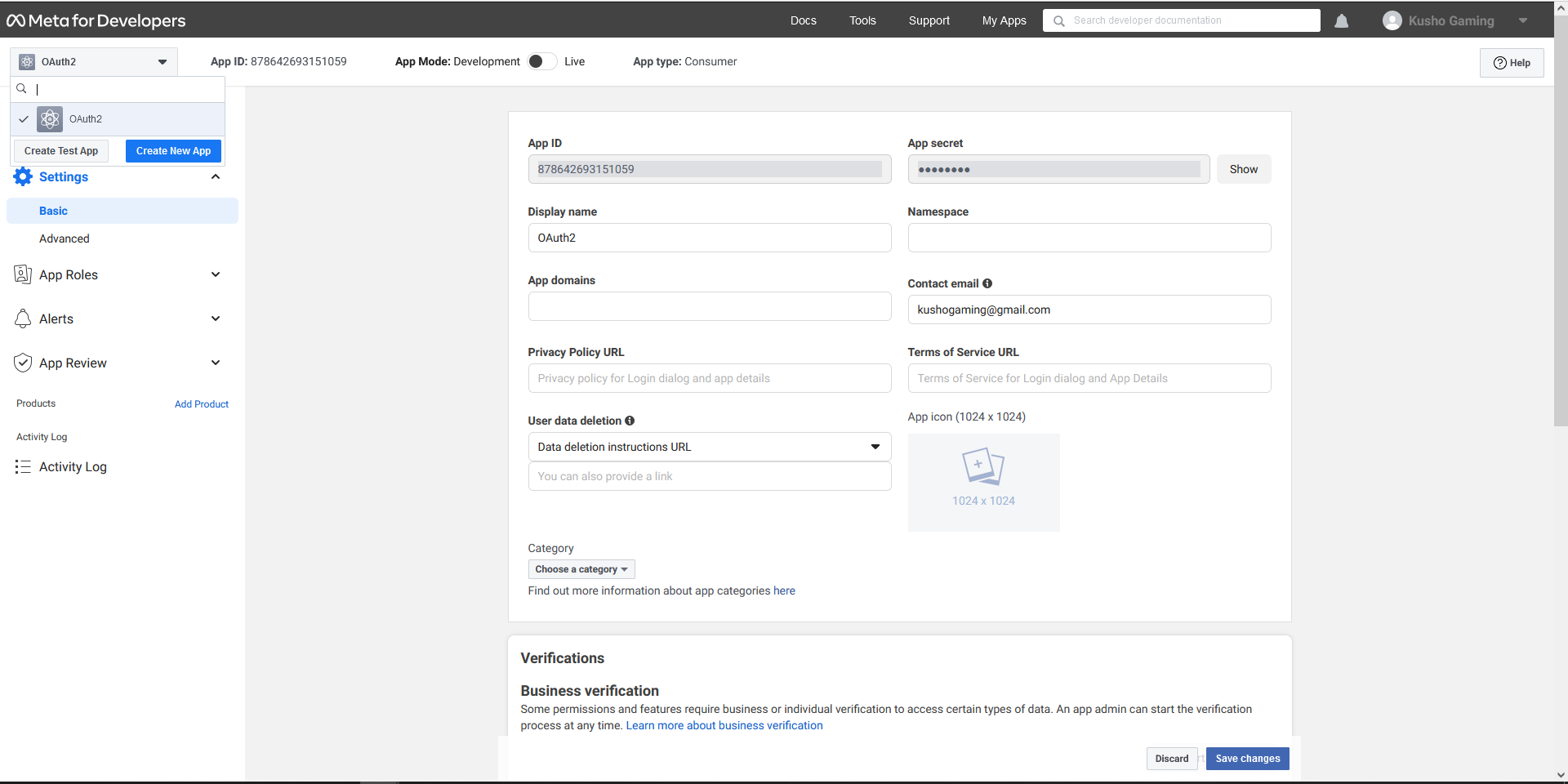




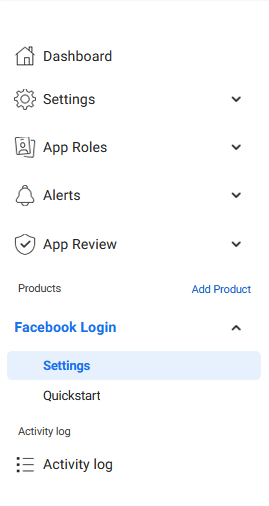
1. Construct the application which implements the selected API in Task 2.

Using Facebook Login API

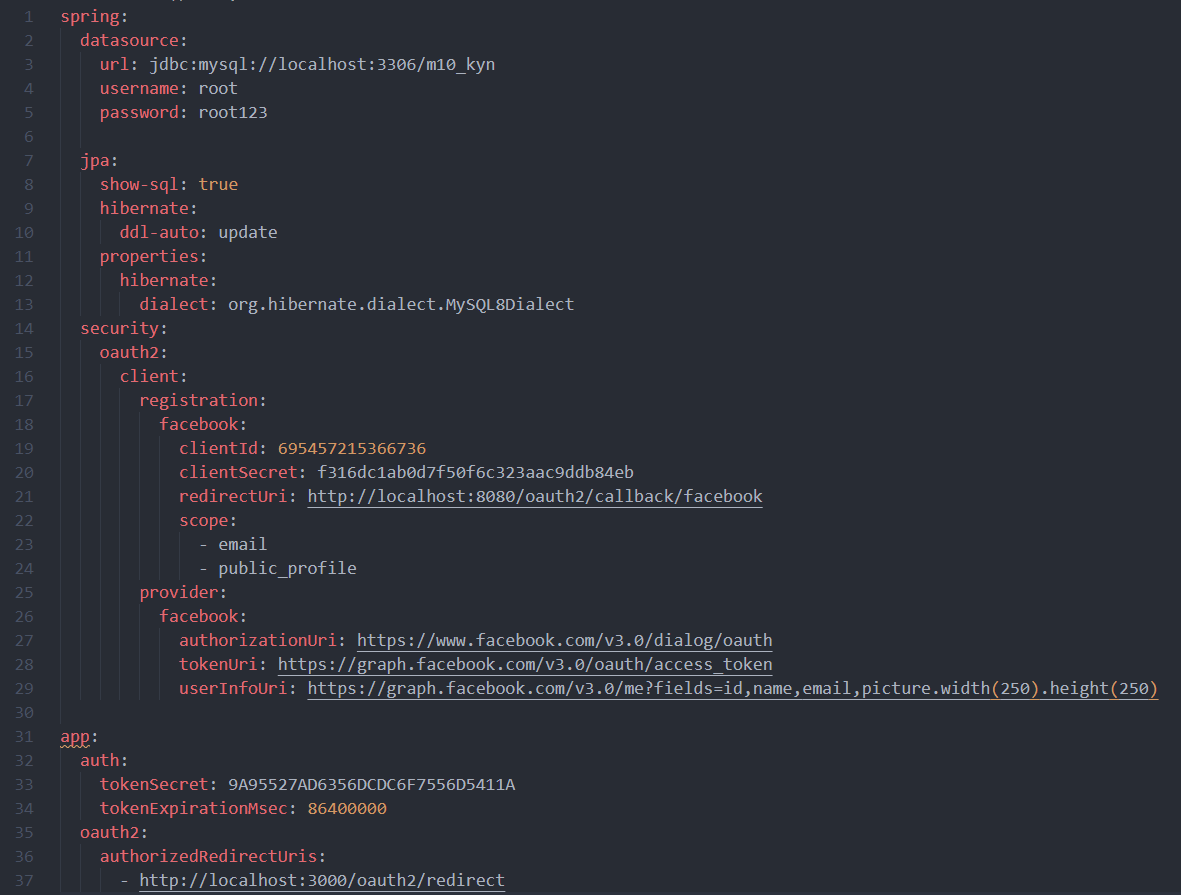
1. Create new developer application on facebook developer website



1. Add Product for Facebook Login



1. Use created App Id and Secret in Application



1. Task 4

Task Statement

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 your Application

What is white box testing?

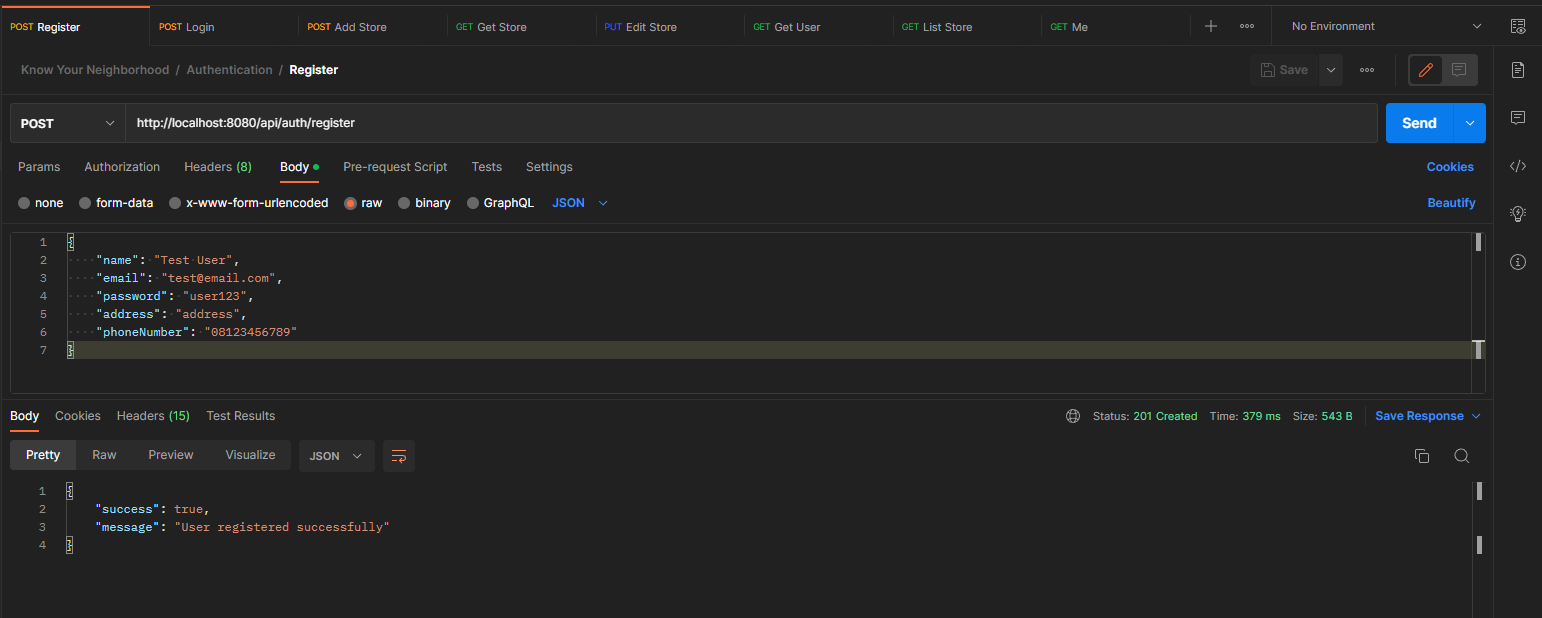
White box testing is a method of testing software that involves examining the internal structure and implementation of the code. It is also known as structural testing or glass box testing.

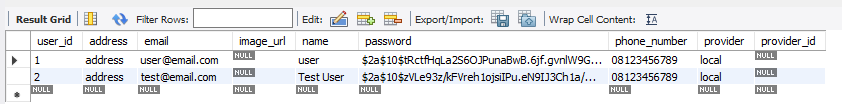
In white box testing, the tester has access to the source code of the software being tested and can use this knowledge to design test cases that exercise specific paths through the code. This can include testing individual functions or methods, as well as testing the interactions between different components of the software.

Testing APIs with Postman

Register

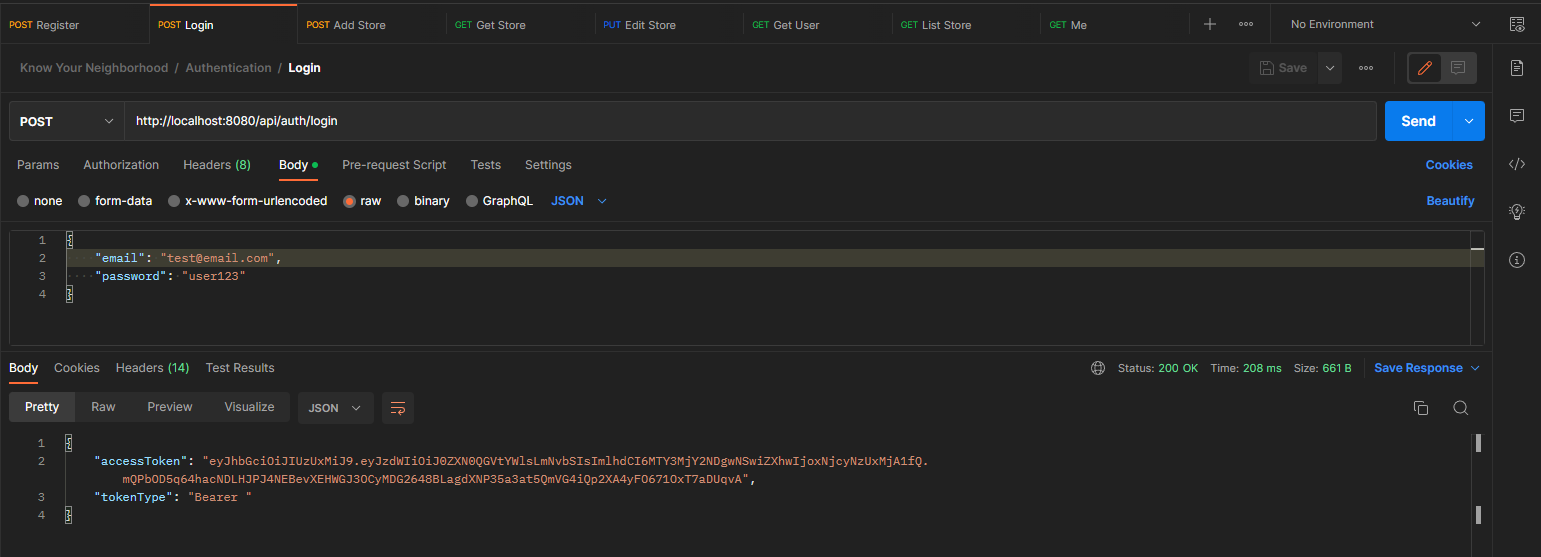
<http://localhost:8080/api/auth/register>





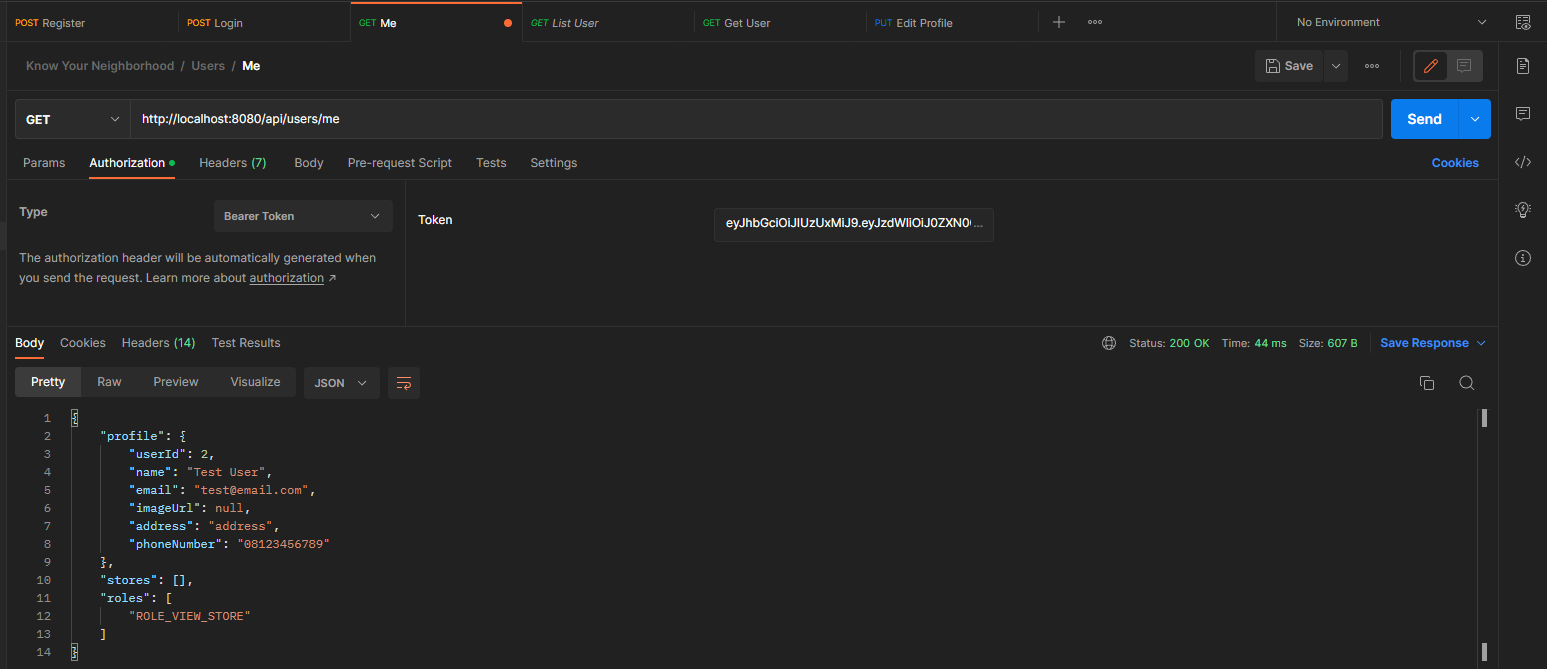
Login

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



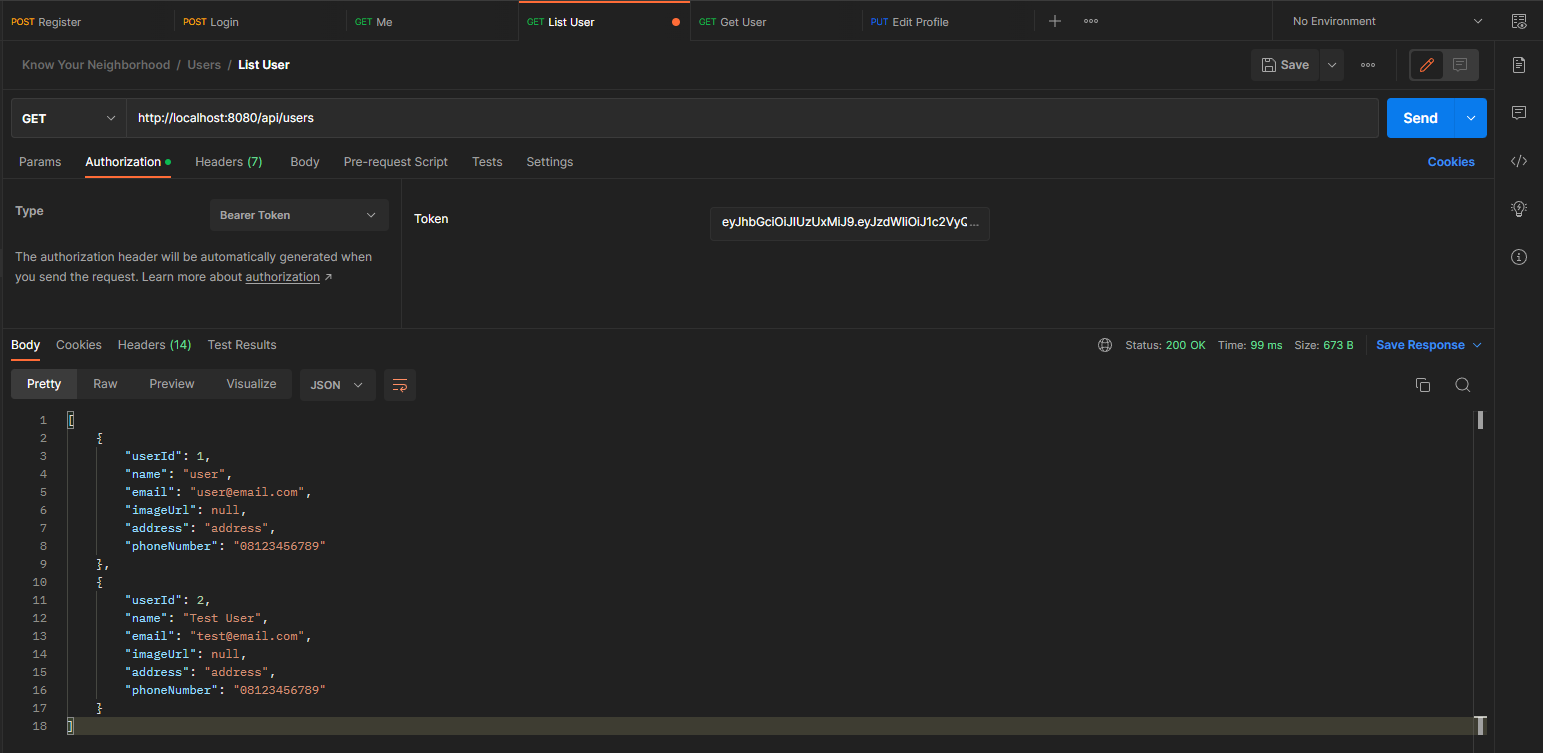
Get User Login

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



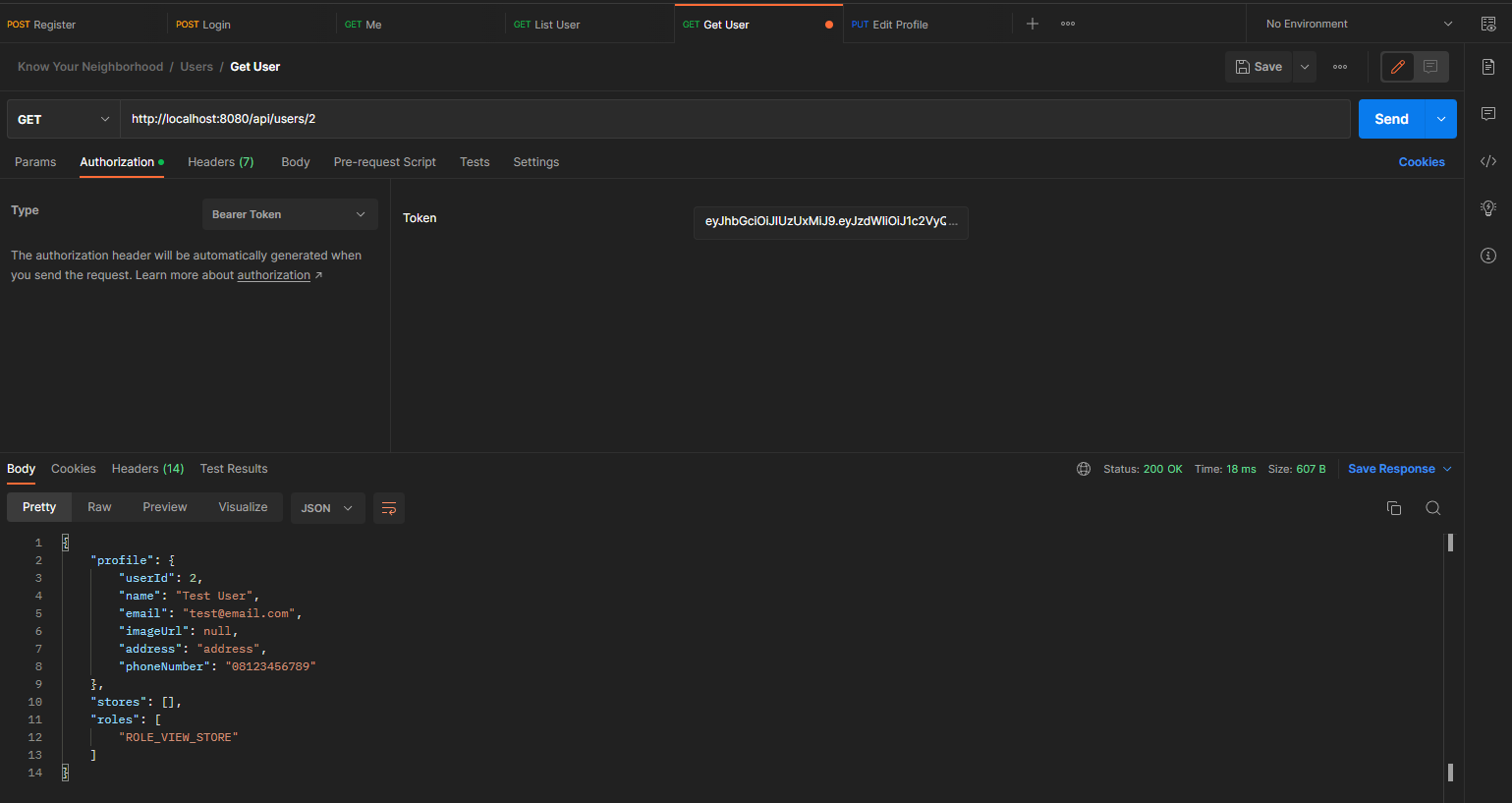
Get List User

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



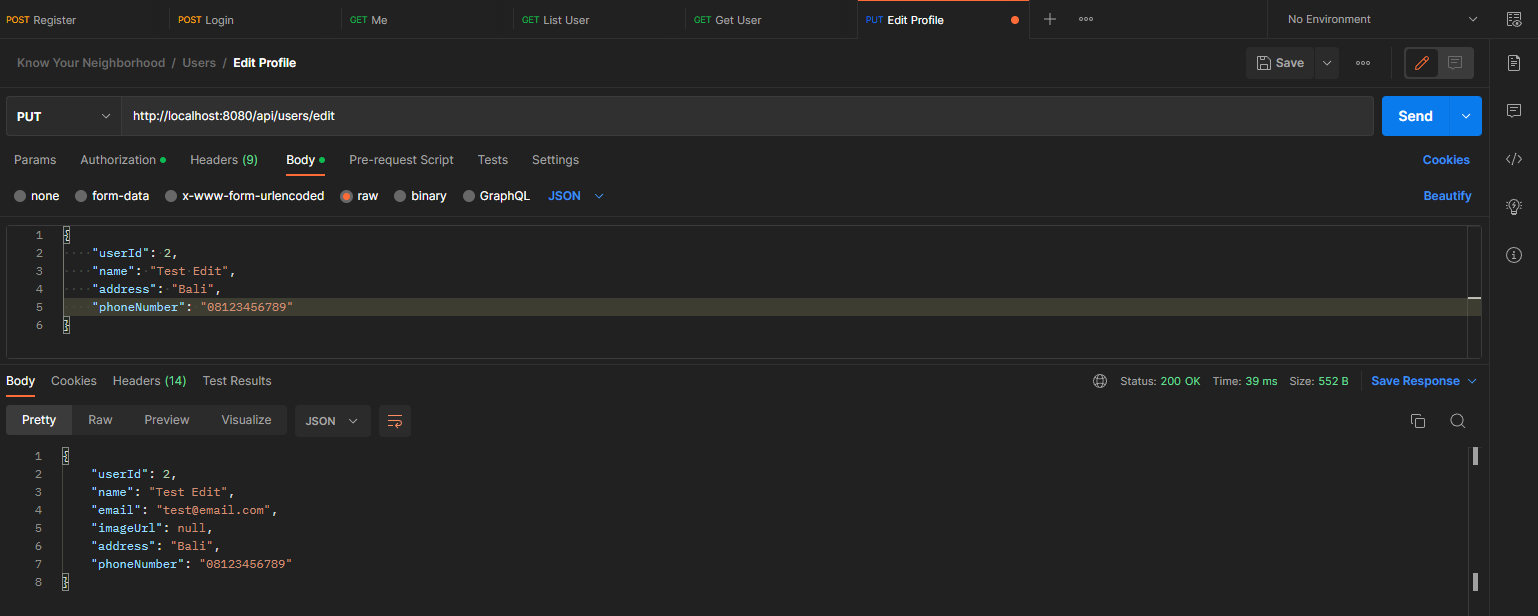
Get User by Id

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

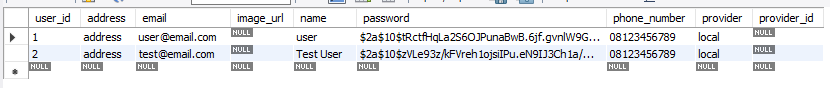


Edit Profile

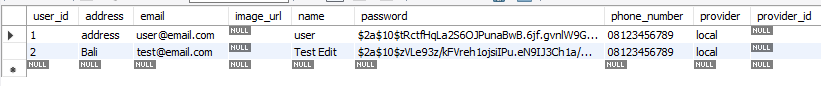
<http://localhost:8080/api/users/edit>



before

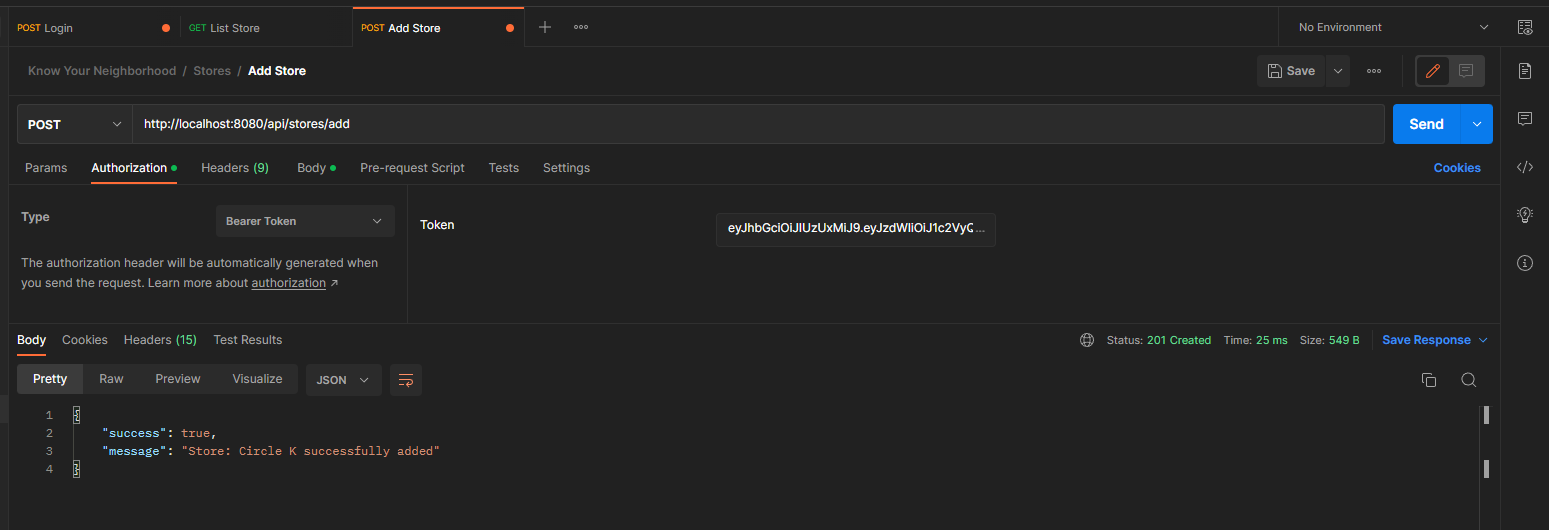


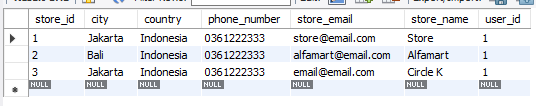
after



Add Store (need ROLE\_ADD\_STORE)

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





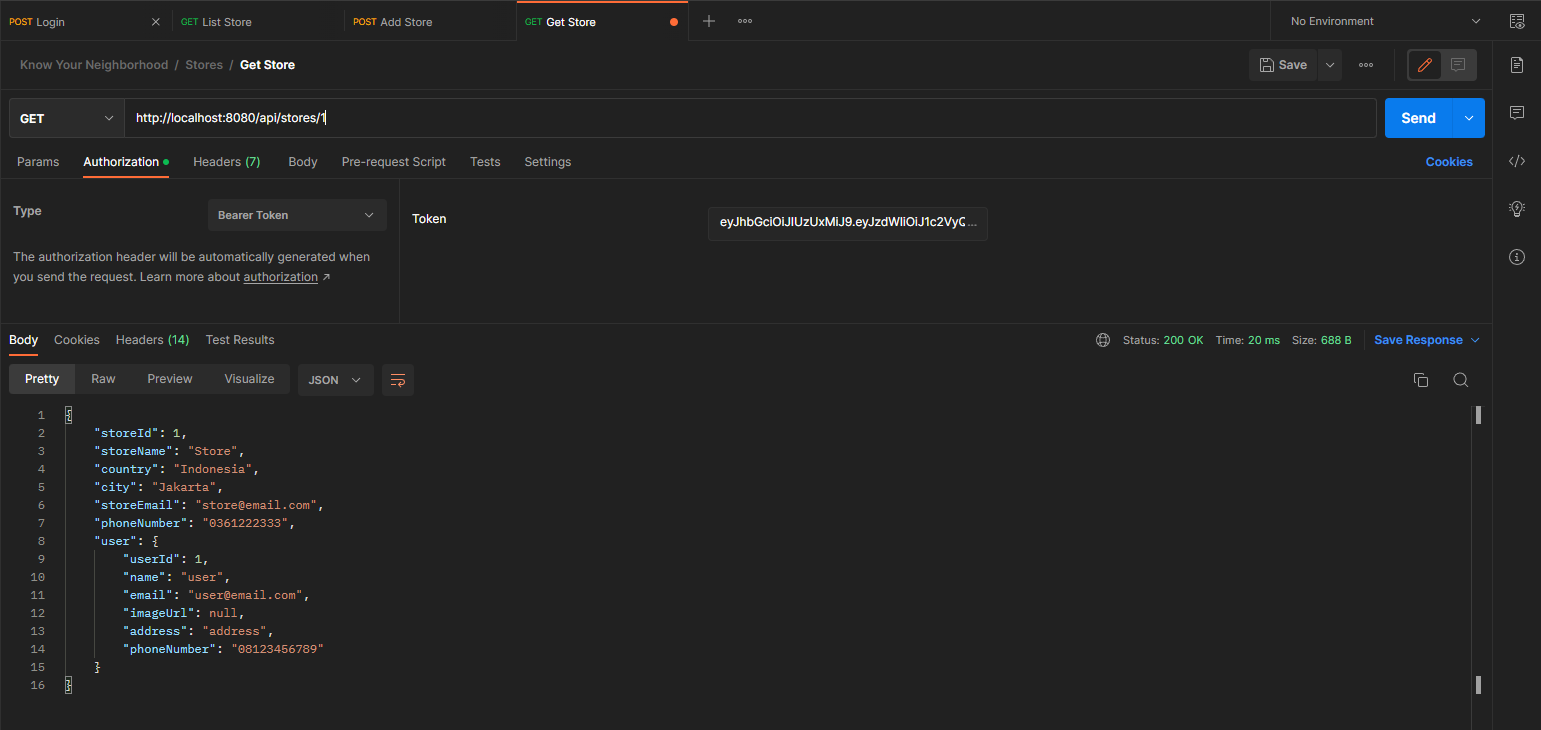
Get List Store

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



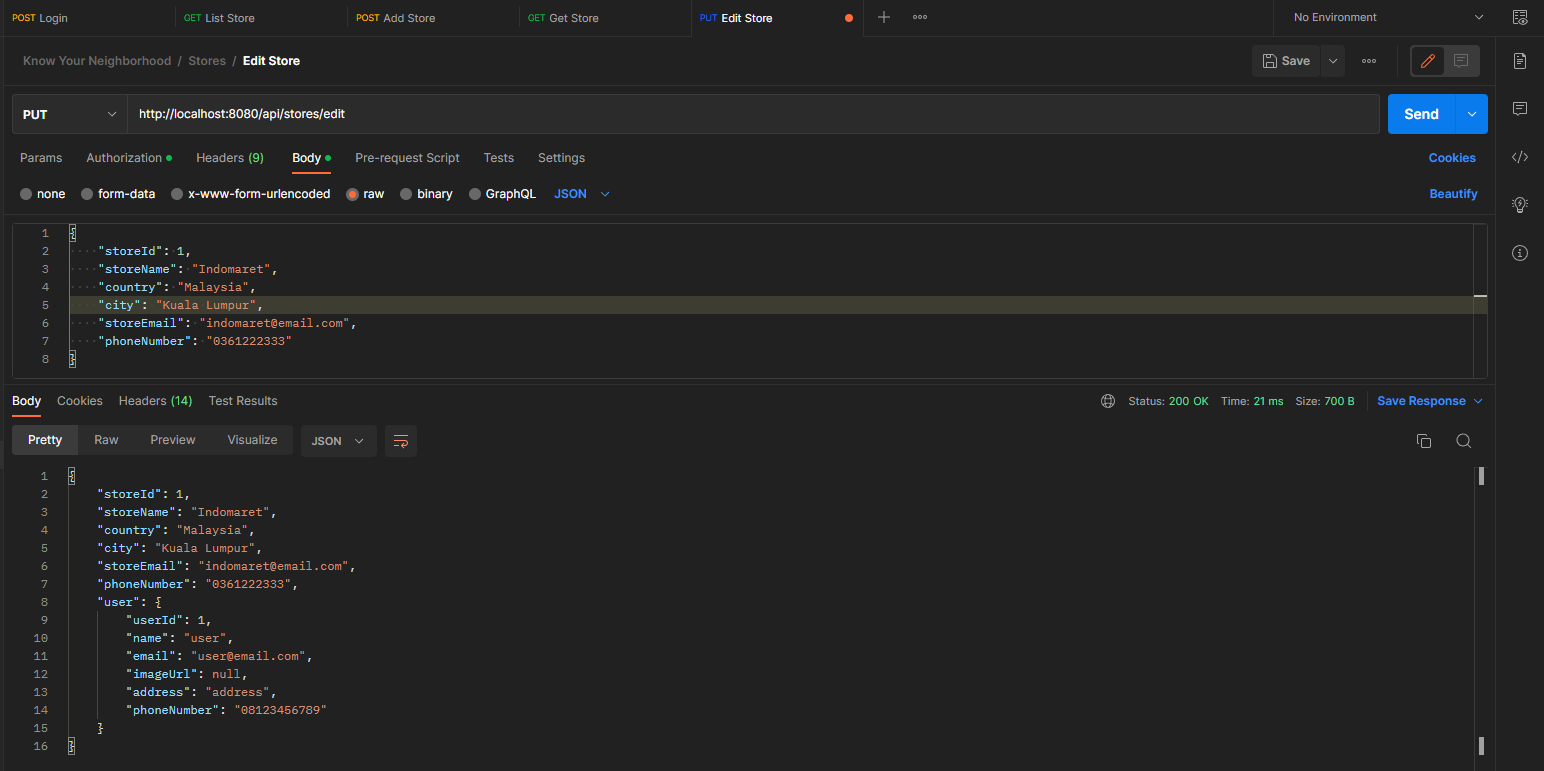
Get Store by Id

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

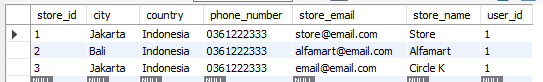


Edit Store (need ROLE\_ADD\_STORE)

<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.

Registration

|  |  |
| --- | --- |
| **Test Scenario** | Registration |
| TS001 |
| **Test Cases** | Registration in Know Your Neighborhood Website |
| TC001 | Validating data that inpputed into registration form |
| TC002 | After user submmiting correct data it should show success message |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Priority** | **Pre-conditions** | **Expected Result** | **Actual Result** | **Final Result** |
| TS001 | TC001 | High | Know Your Neighborhood Website hosted on server | If user enter blank data It should throw an error | As expected | Pass |
| TC002 | High | Show success message after submitting data | As expected | Pass |

**Test Data Table**

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Data** |
| TS001 | TC001 | email = “ ” (blank or space)  Password = user123  Name = gustut  Address = Bali  Phone number = 08123456789 |
| TC002 | email = yoghantara@email.com  Password = user123  Name = gustut  Address = Bali  Phone number = 08123456789 |

**Test Evidences**

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

Login

|  |  |
| --- | --- |
| **Test Scenario** | Login |
| TS001 |
| **Test Cases** | Login in Know Your Neighborhood Website |
| TC001 | Login with local account |
| TC002 | Login with facebook API |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Priority** | **Pre-conditions** | **Expected Result** | **Actual Result** | **Final Result** |
| TS001 | TC001 | High | Know Your Neighborhood Website hosted on server | After user entering valid data, user should be redirected to their profile page | As expected | Pass |
| TC002 | High | After user entering valid data, user should be redirected to their profile page | As expected | Pass |

**Test Data Table**

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Data** |
| TS001 | TC001 | Email = [user@email.com](mailto:user@email.com)  Password = user123 |
| TC002 | “Valid facebook account” |

**Test Evidences**

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

1. Task 5

Task Statement

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.

Strength of REST APIs

* They are simple to use and understand, making them accessible to a wide range of developers.
* They use HTTP standard methods (e.g. GET, POST, PUT, DELETE) to transmit data, making them easy to integrate with other systems and tools.
* They are lightweight, making them suitable for use with a wide range of devices and networks.
* They are stateless, meaning that each request stands on its own, without needing to maintain a session or connection.
* They are platform-independent, so they can be used with a variety of programming languages and frameworks.

Weaknesses of RESP APIs

* They can be less secure than other types of APIs, as they rely on the underlying transport protocol (HTTP) to provide security.
* They can be prone to errors if not implemented properly, as they rely on clients to provide correct input.
* They can be less efficient than other types of APIs, as they require additional overhead (e.g. parsing JSON or XML) to process data.
* Some use cases may require more complex features than REST can provide.

1. Provide data security report of your developed application.

Spring Security

This application use spring security to secure its application. There are several reasons to use Spring Security:

* It is highly customizable: Spring Security can be easily configured to meet the specific security needs of your application.
* It integrates well with other Spring projects: In this scenario we are using Spring framework so Spring Security can be easily integrated into existing infrastructure.

Authentication with JSON Web Token (JWT)

Authentication with JSON Web Token (JWT) is a popular technique for securing web applications. There are several reasons why this project chooses to use JWT for authentication:

* JWTs are self-contained: A JWT contains all the information necessary to authenticate a user in a single, compact token. This eliminates the need to store authentication information in a session or on the server, which can simplify the implementation of the application.
* JWTs are secure: JWTs are signed with a secret key, which makes it difficult for attackers to forge them or tamper with their contents.
* JWTs are widely supported: JWT is a standard that is supported by a wide range of platforms and libraries, making it easy to integrate into your application.

Security with OAuth2

OAuth2 allows users to grant third-party applications access to their resources without sharing their login credentials. This can help to prevent unauthorized access to user accounts and data.

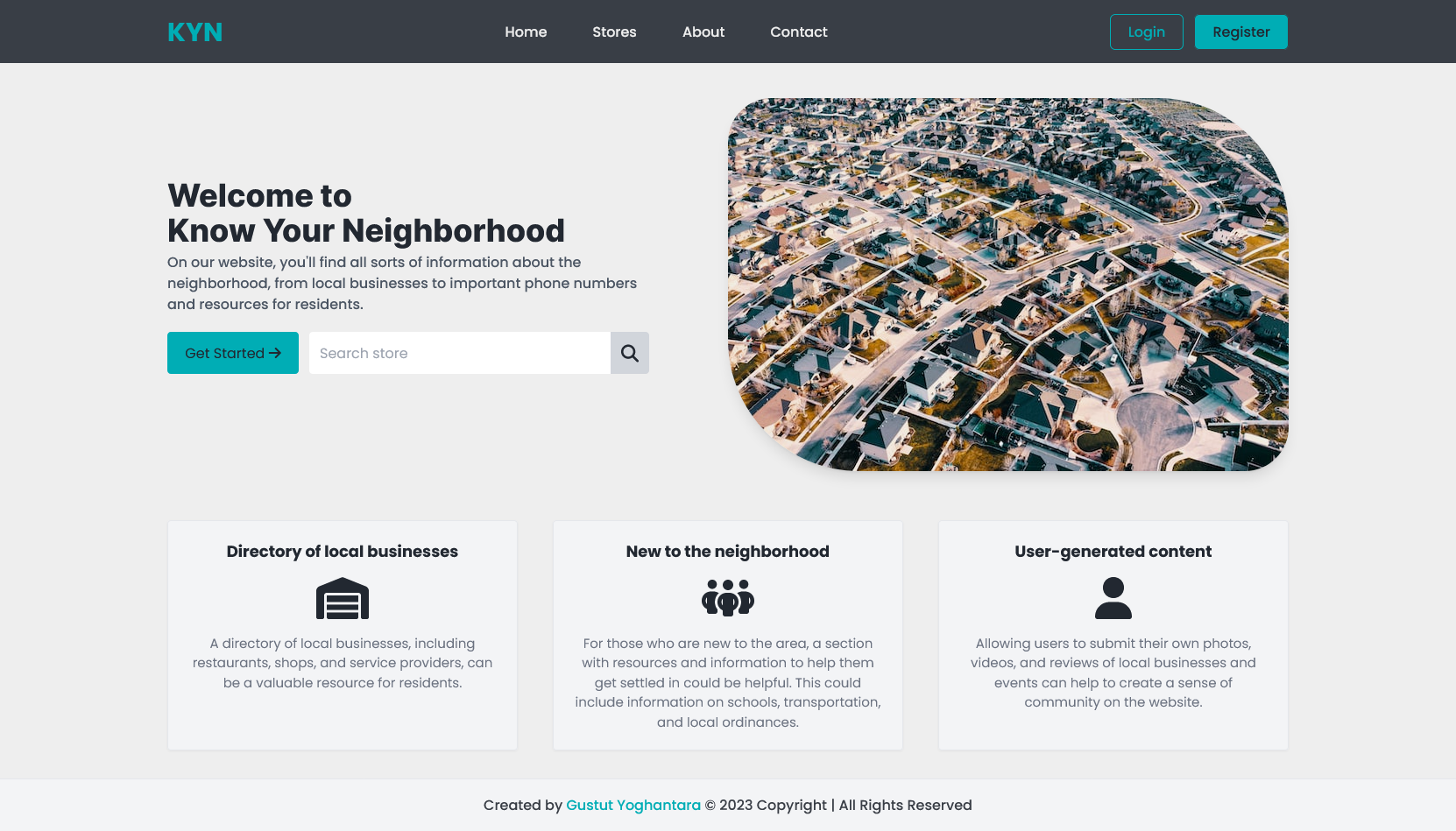
1. Task 6

Task Statement:

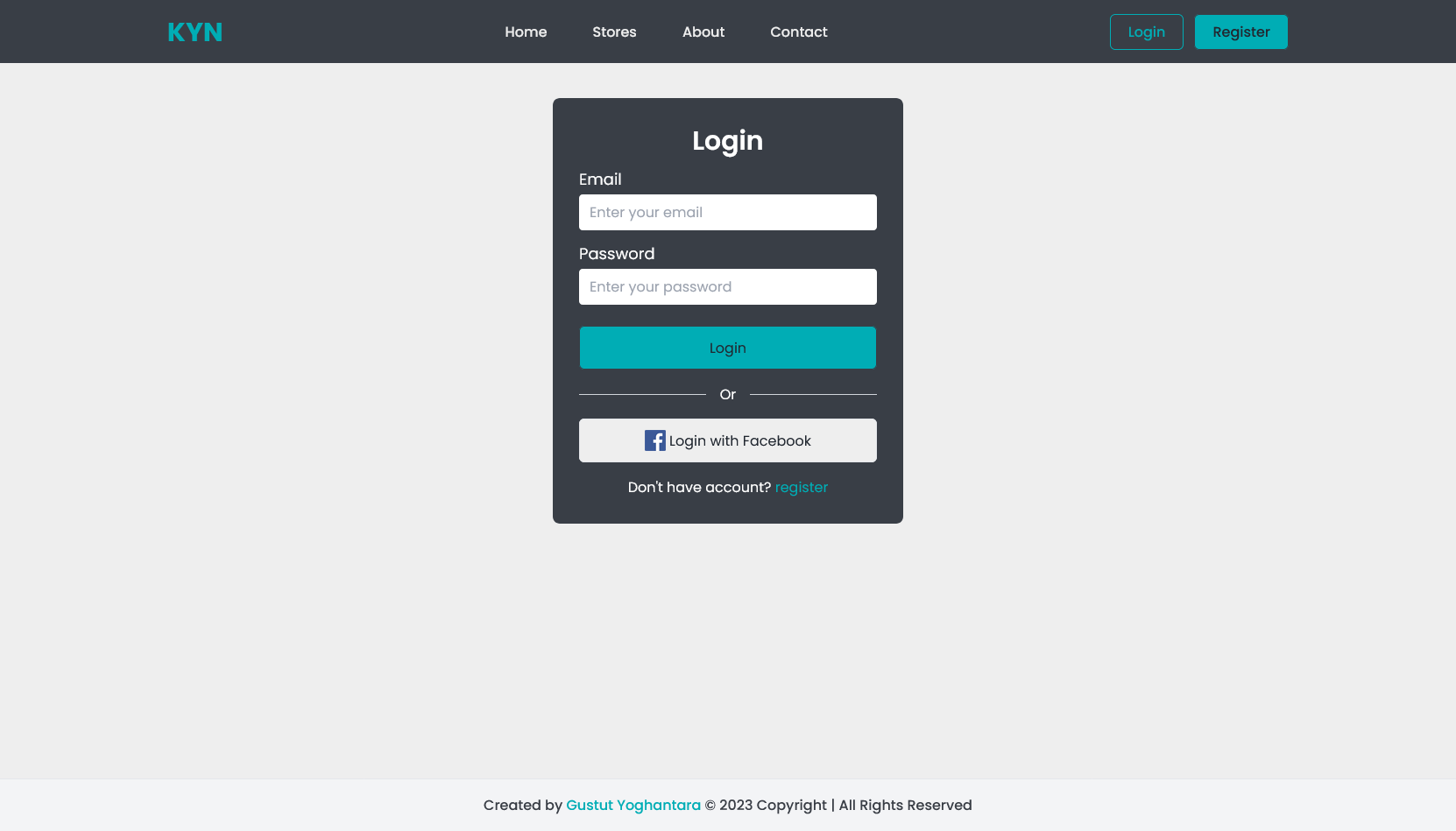
1. Provide screen capture of developed application using APIs in Project Presentation

Solution:

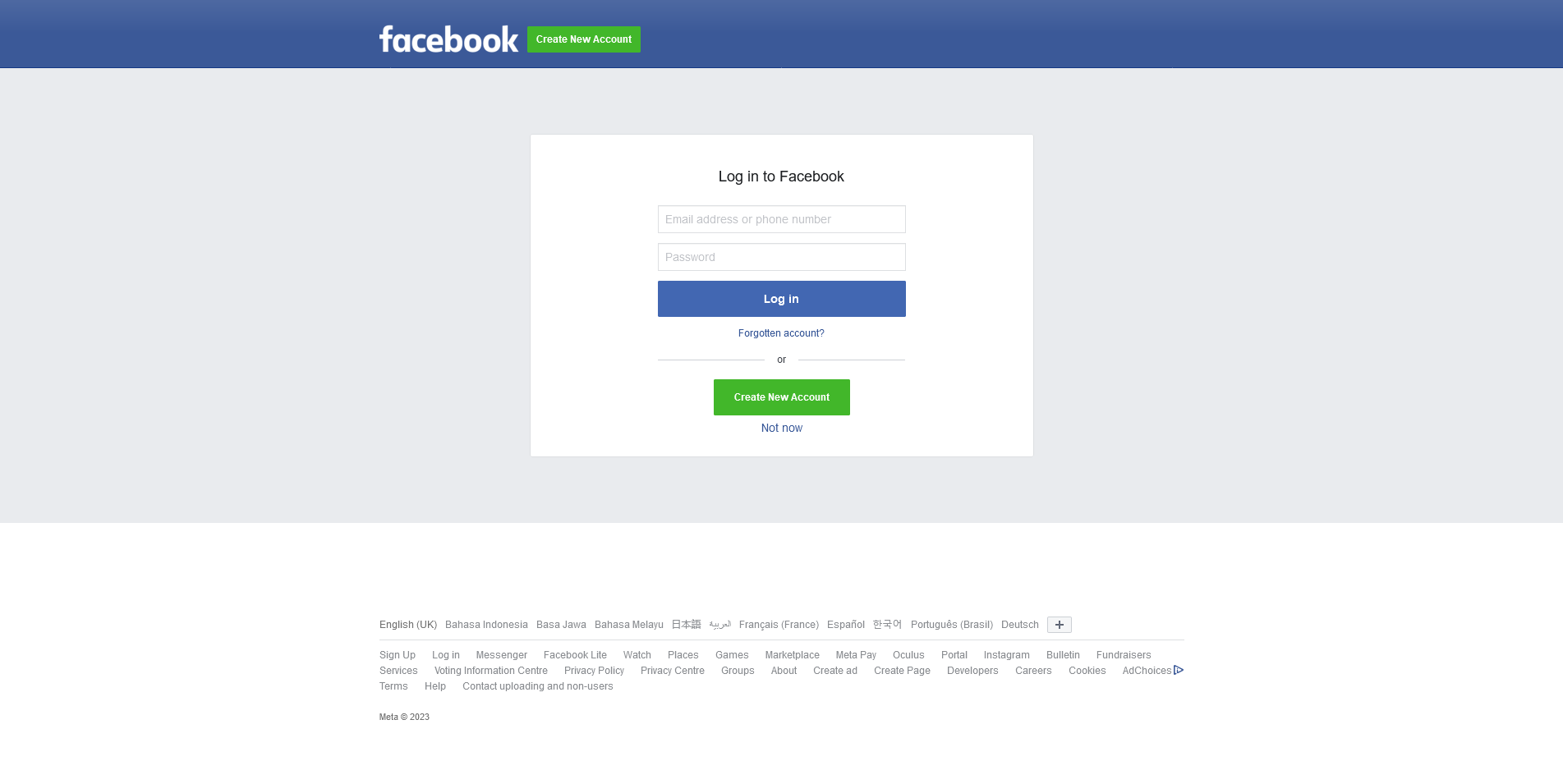
Home Page



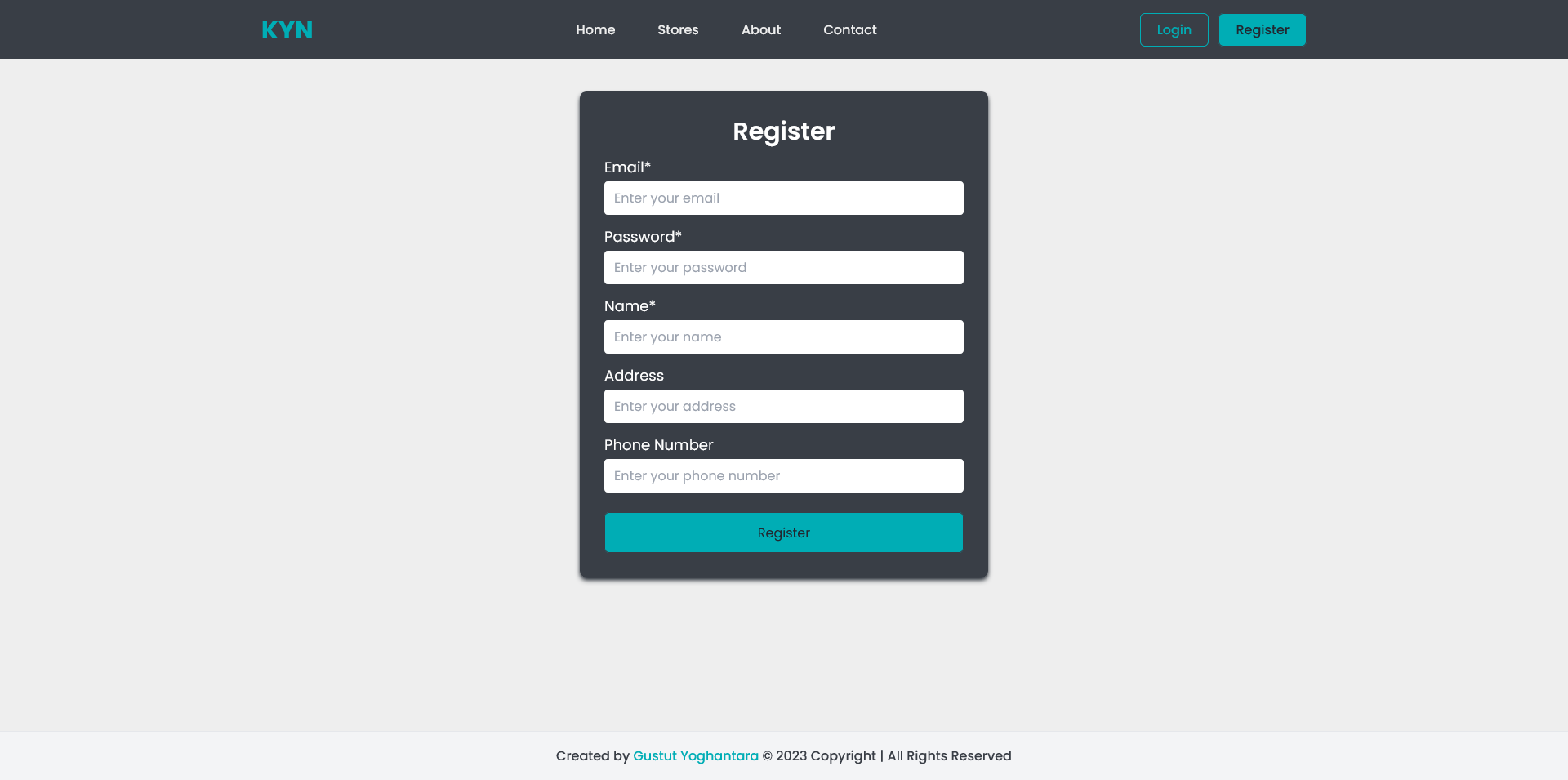
Login Page



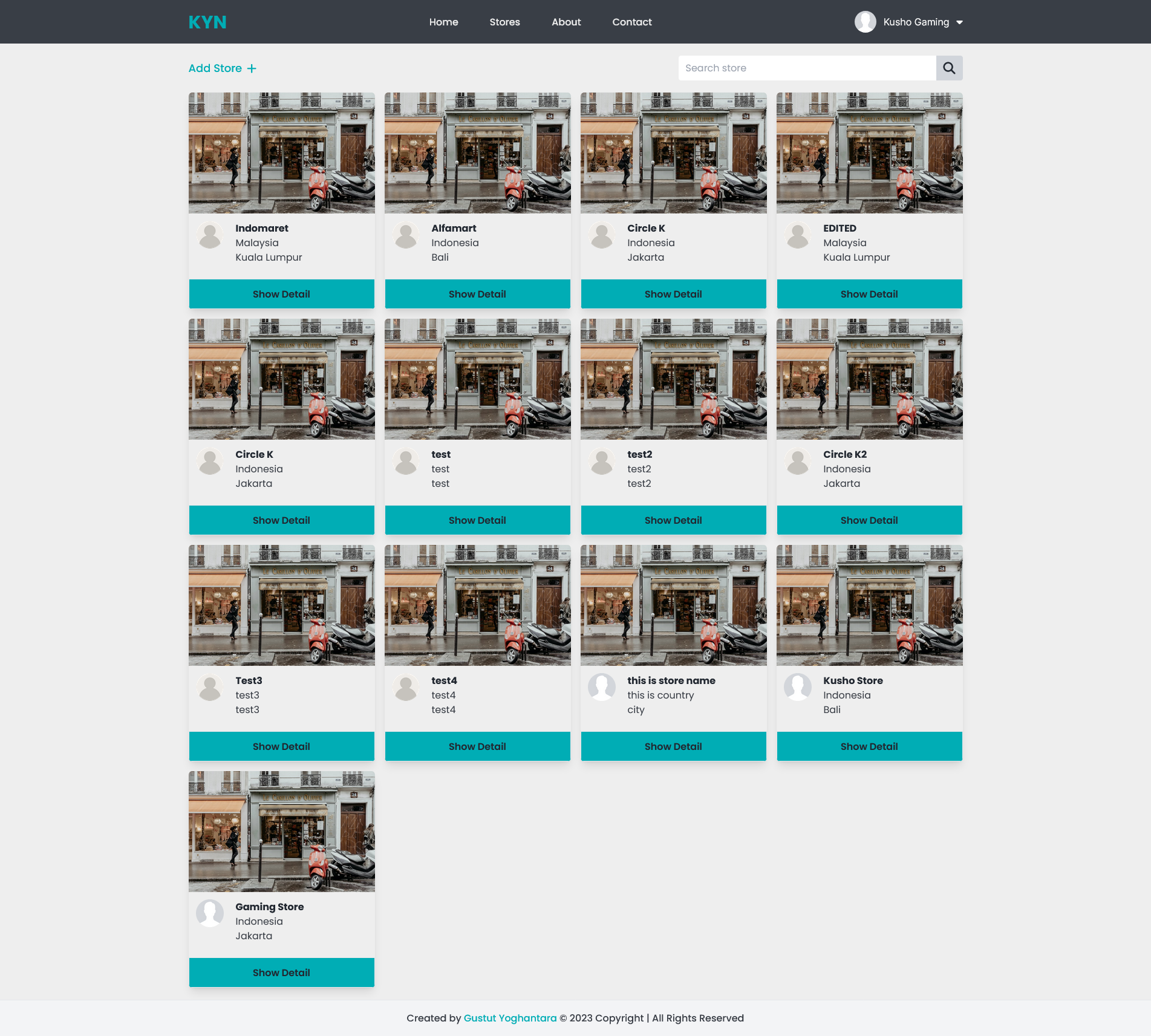
Facebook Login



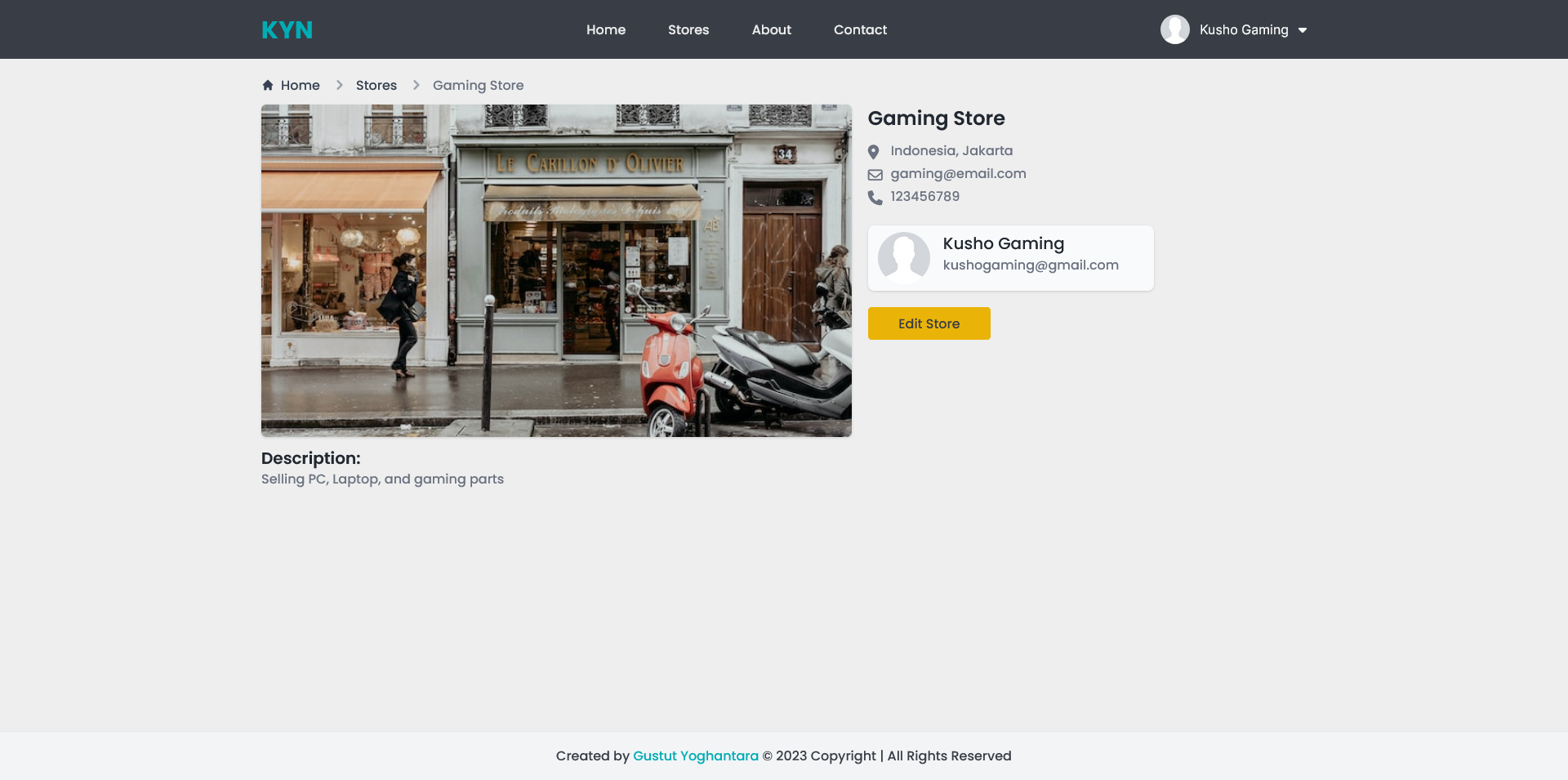
Register



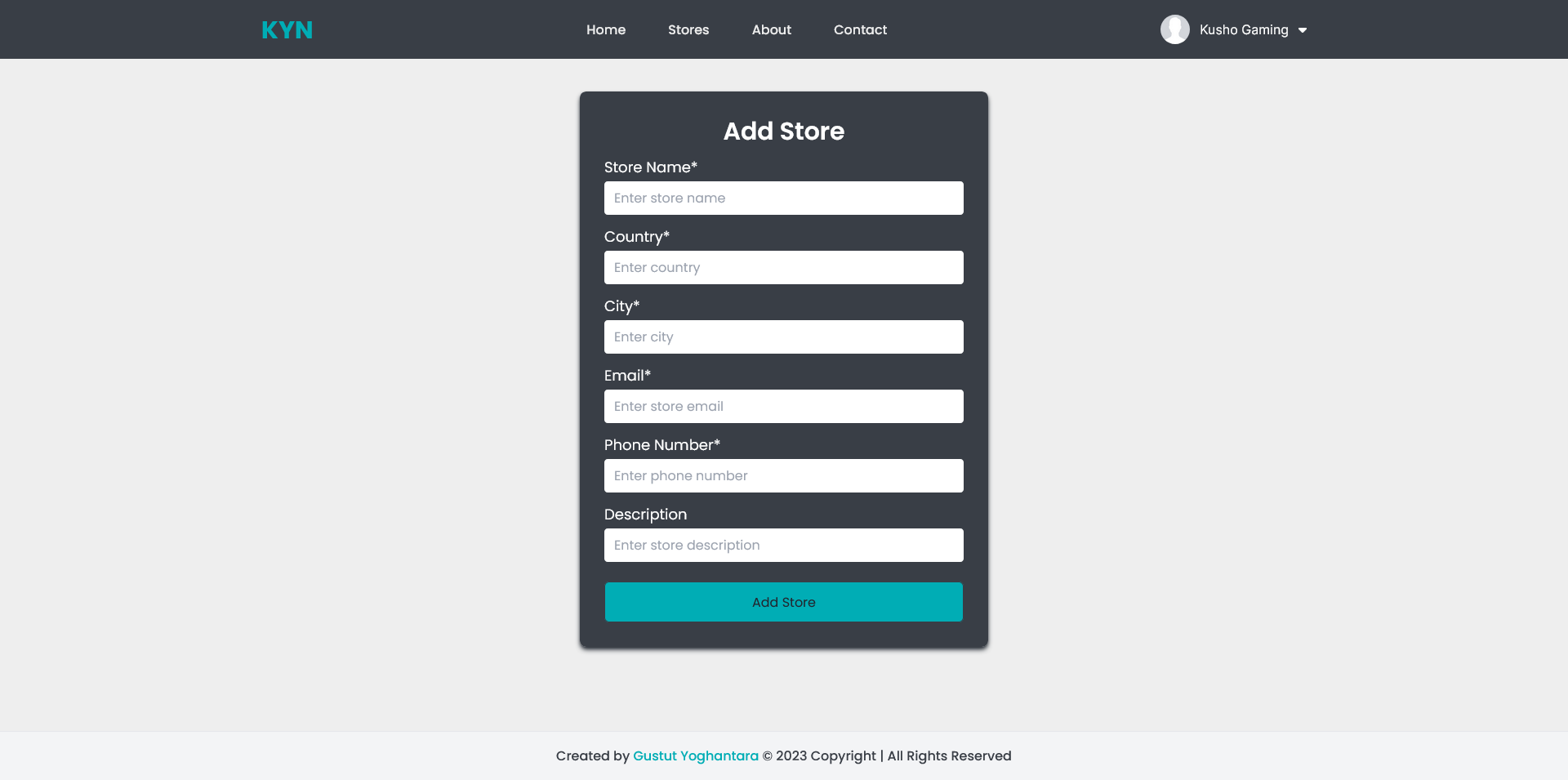
Stores



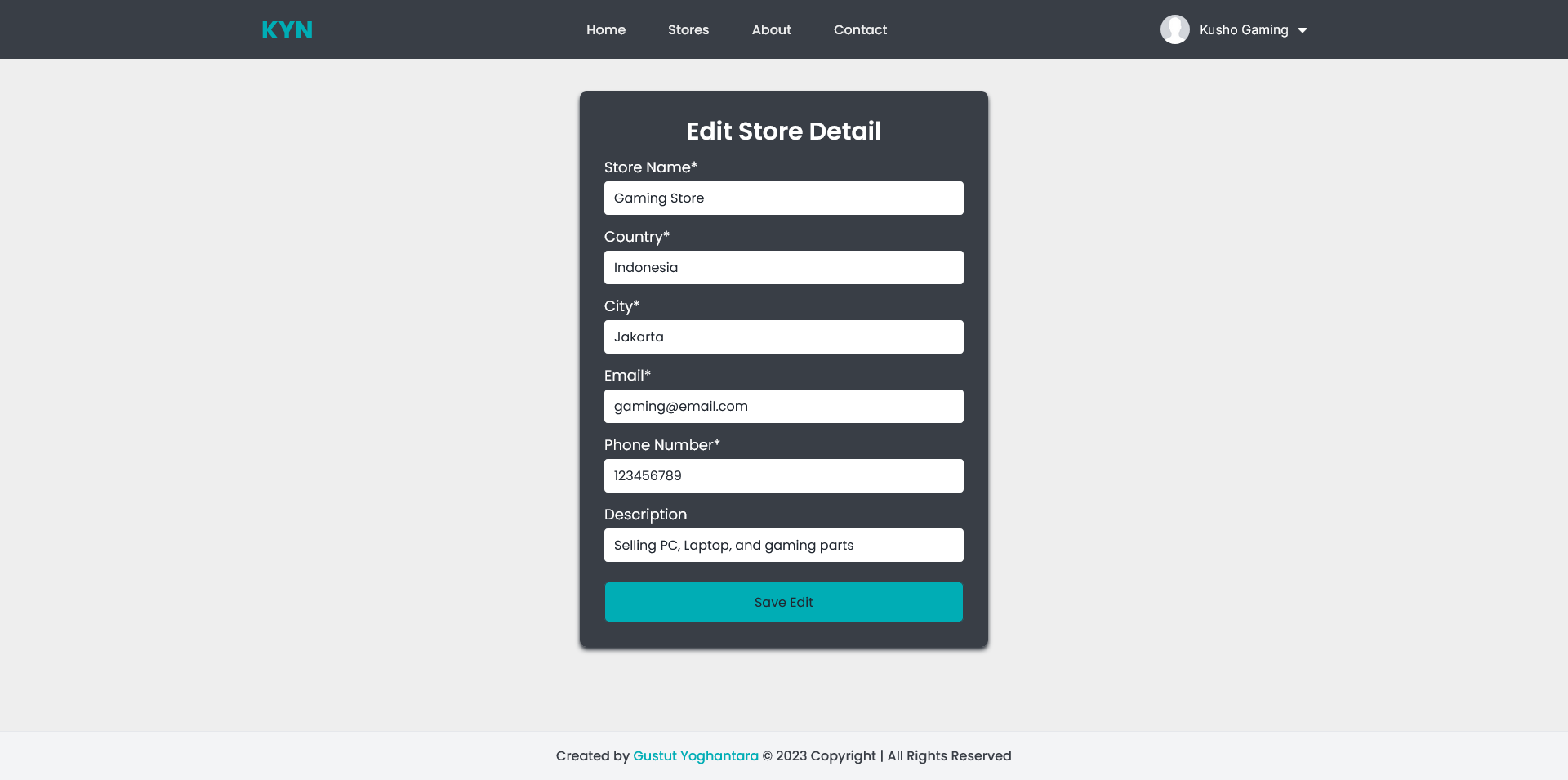
Store Detail



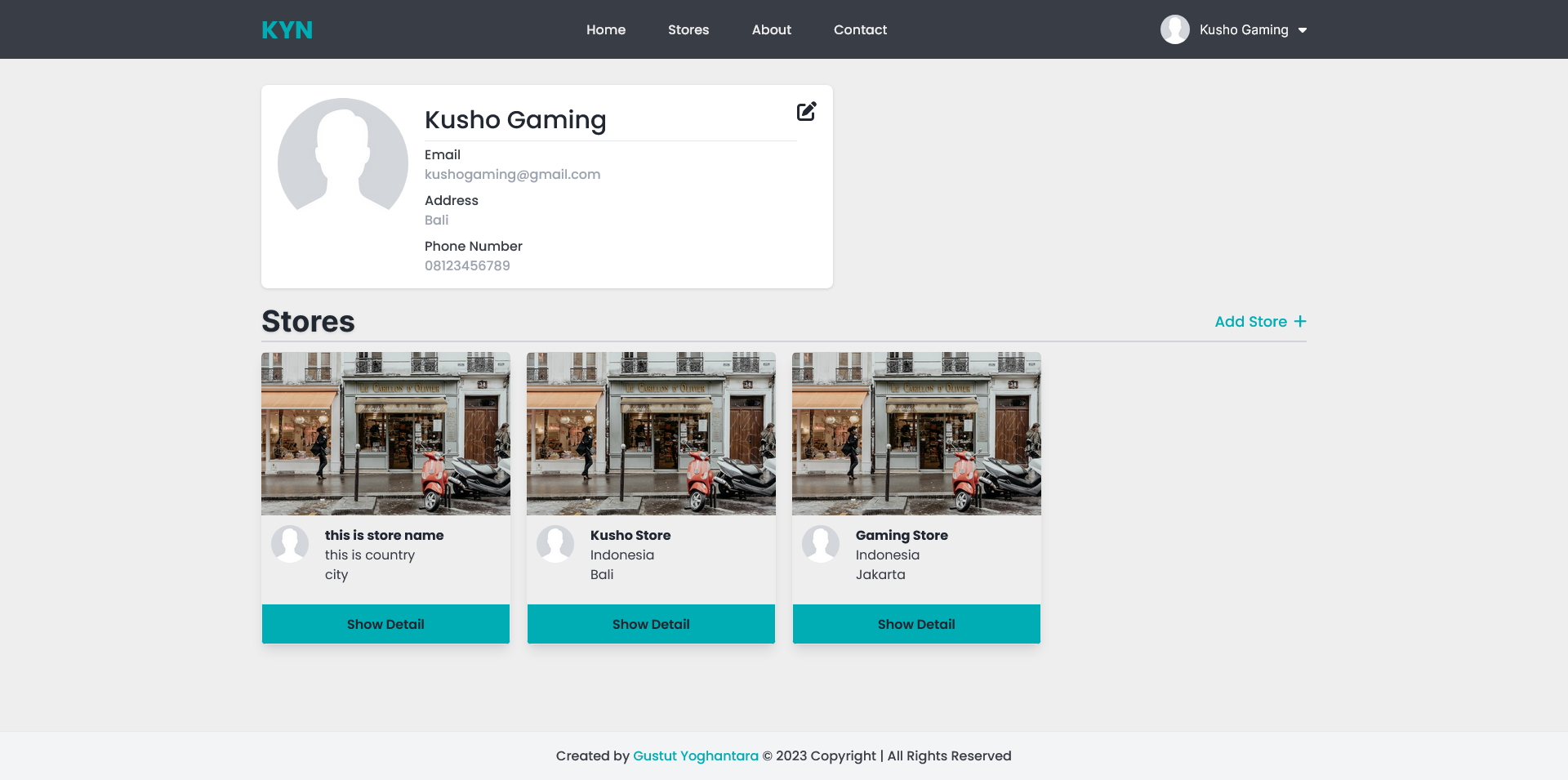
Add Store



Edit Store



Profile Page



Edit Profile

