LITHAN

Application Programming Interface

Module Project

Module: Application Programming Interface

Start Date	**: 2 January 2023**
End Date	**: 11 January 2023**
Submission Date	**: 30 January 2023**

Learner Name	**: Chathushi Jayarathna**
Enrollment ID	**:**
Presentation Date	**:**

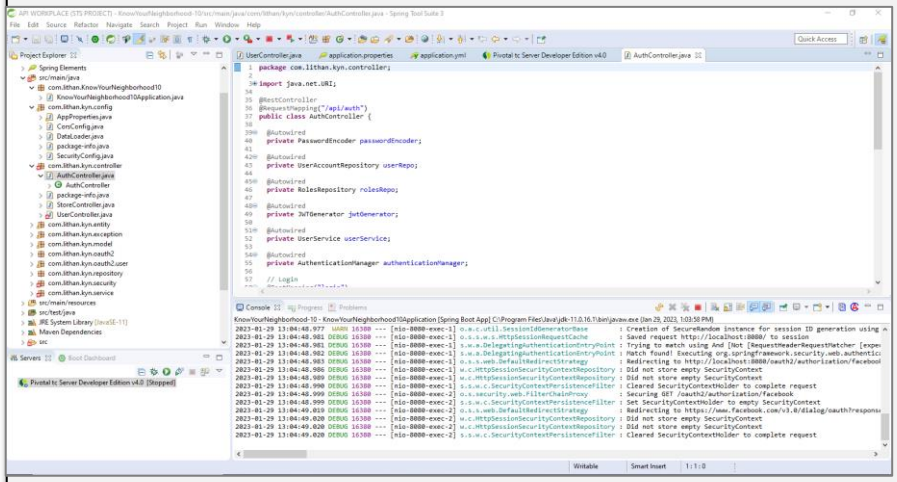
Contents

LITHAN

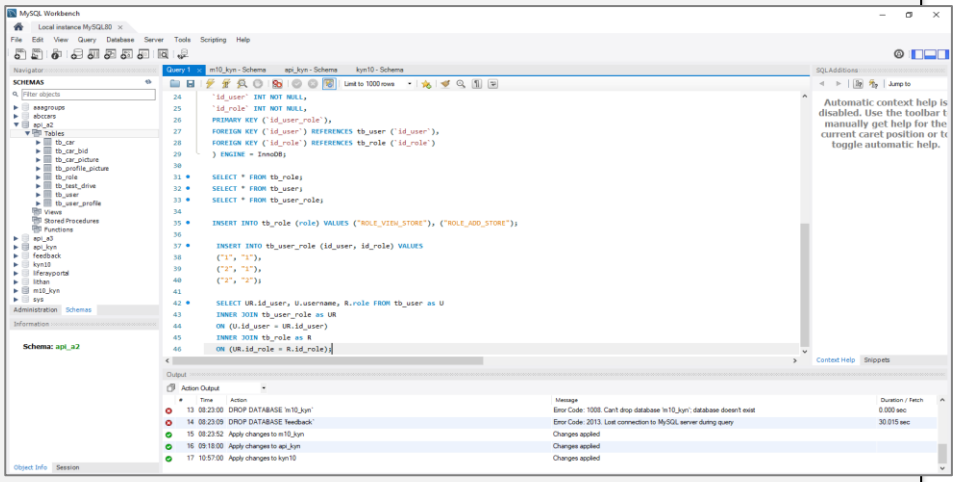
1. List of Tools used

LITHAN

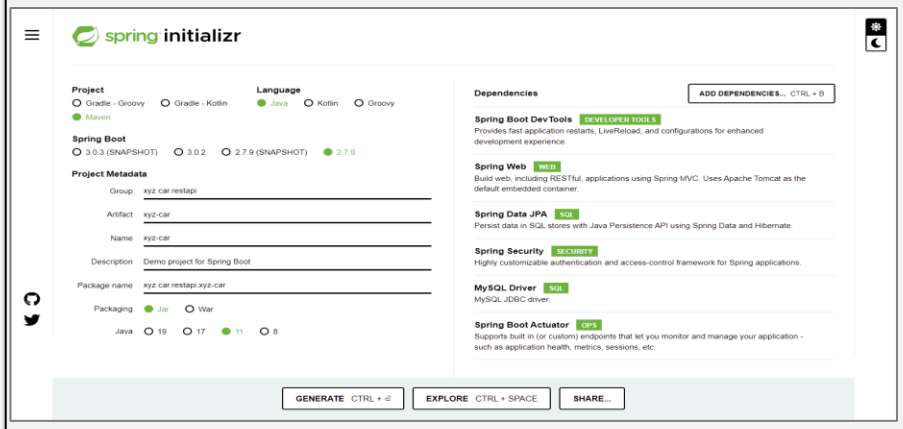
Sprint Boot



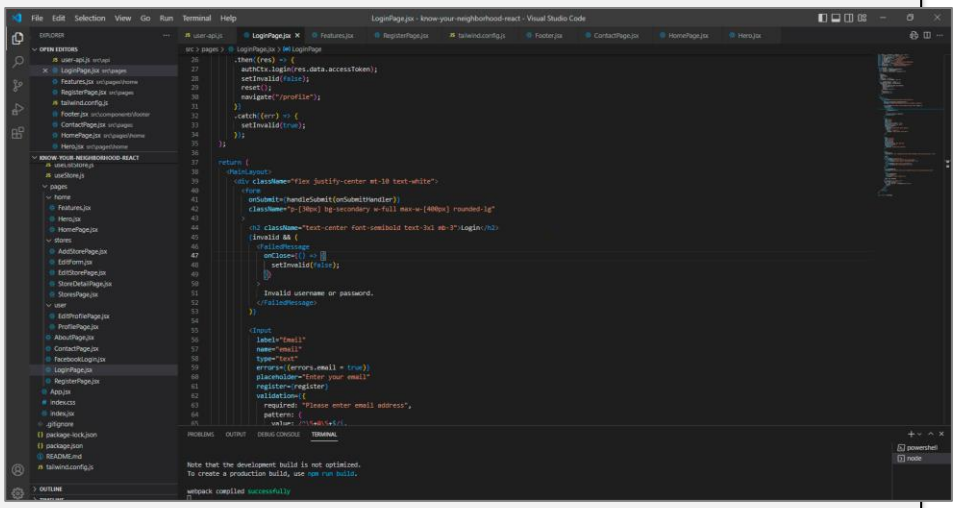
MySQL Workbench



Spring Initializr



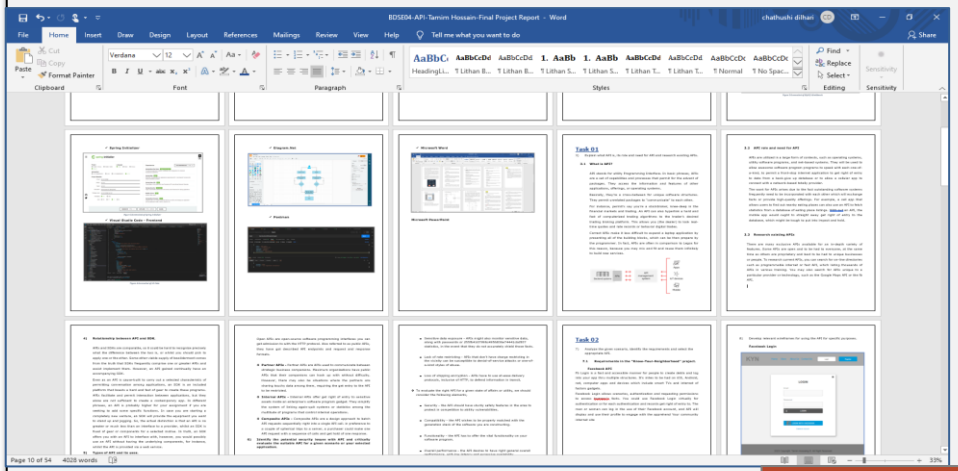
Visual Studio Code



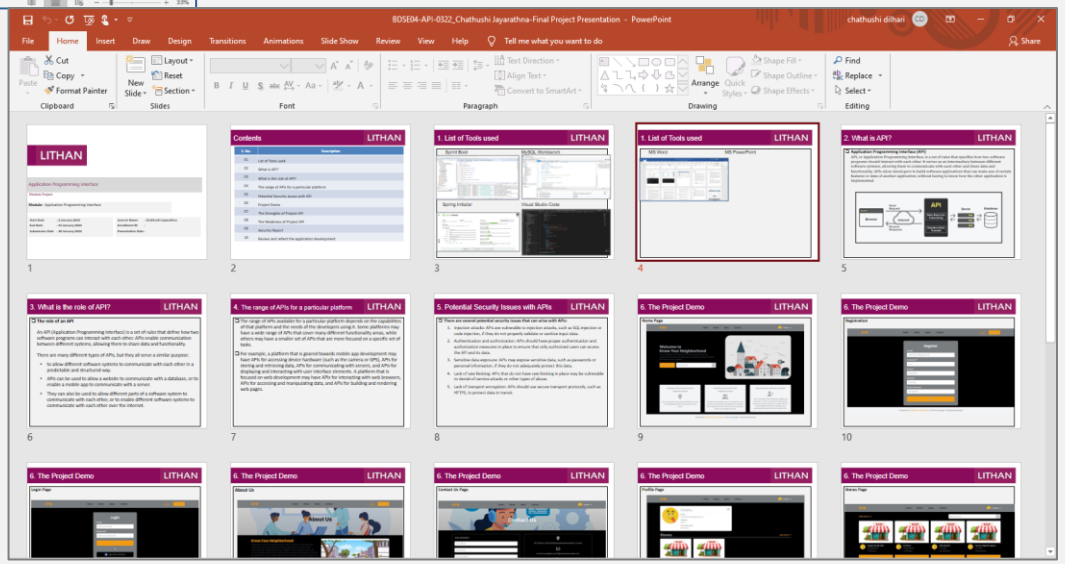
1. List of Tools used

LITHAN

MS Word



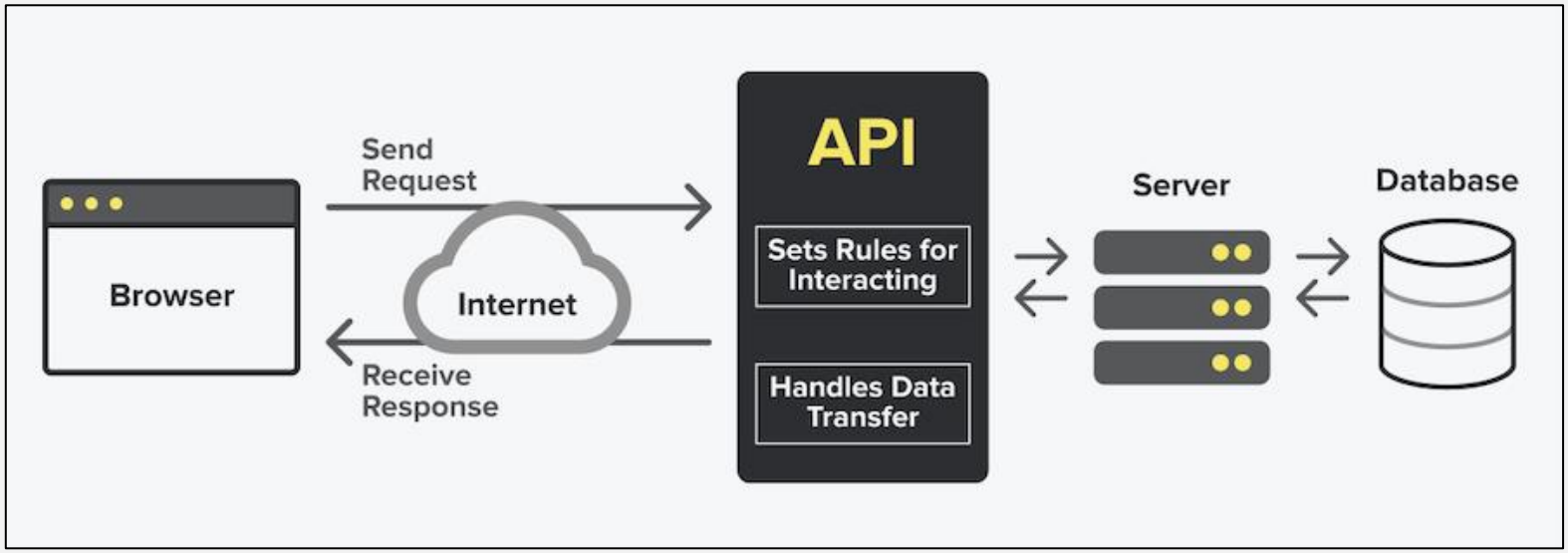
MS PowerPoint



2. What is API?

LITHAN

- **Application Programming Interface (API)**
  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.



3. What is the role of API?

LITHAN

❑ **The role of an API**

An API (Application Programming Interface) is a set of rules that define how two software programs can interact with each other. APIs enable communication between different systems, allowing them to share data and functionality.

There are many different types of APIs, but they all serve a similar purpose:

- to allow different software systems to communicate with each other in a predictable and structured way.
- APIs can be used to allow a website to communicate with a database, or to enable a mobile app to communicate with a server.
- They can also be used to allow different parts of a software system to communicate with each other, or to enable different software systems to communicate with each other over the internet.

4. The range of APIs for a particular platform

LITHAN

- The range of APIs available for a particular platform depends on the capabilities of that platform and the needs of the developers using it. Some platforms may have a wide range of APIs that cover many different functionality areas, while others may have a smaller set of APIs that are more focused on a specific set of tasks.
- For example, a platform that is geared towards mobile app development may have APIs for accessing device hardware (such as the camera or GPS), APIs for storing and retrieving data, APIs for communicating with servers, and APIs for displaying and interacting with user interface elements. A platform that is focused on web development may have APIs for interacting with web browsers, APIs for accessing and manipulating data, and APIs for building and rendering web pages.

5. Potential Security Issues with APIs

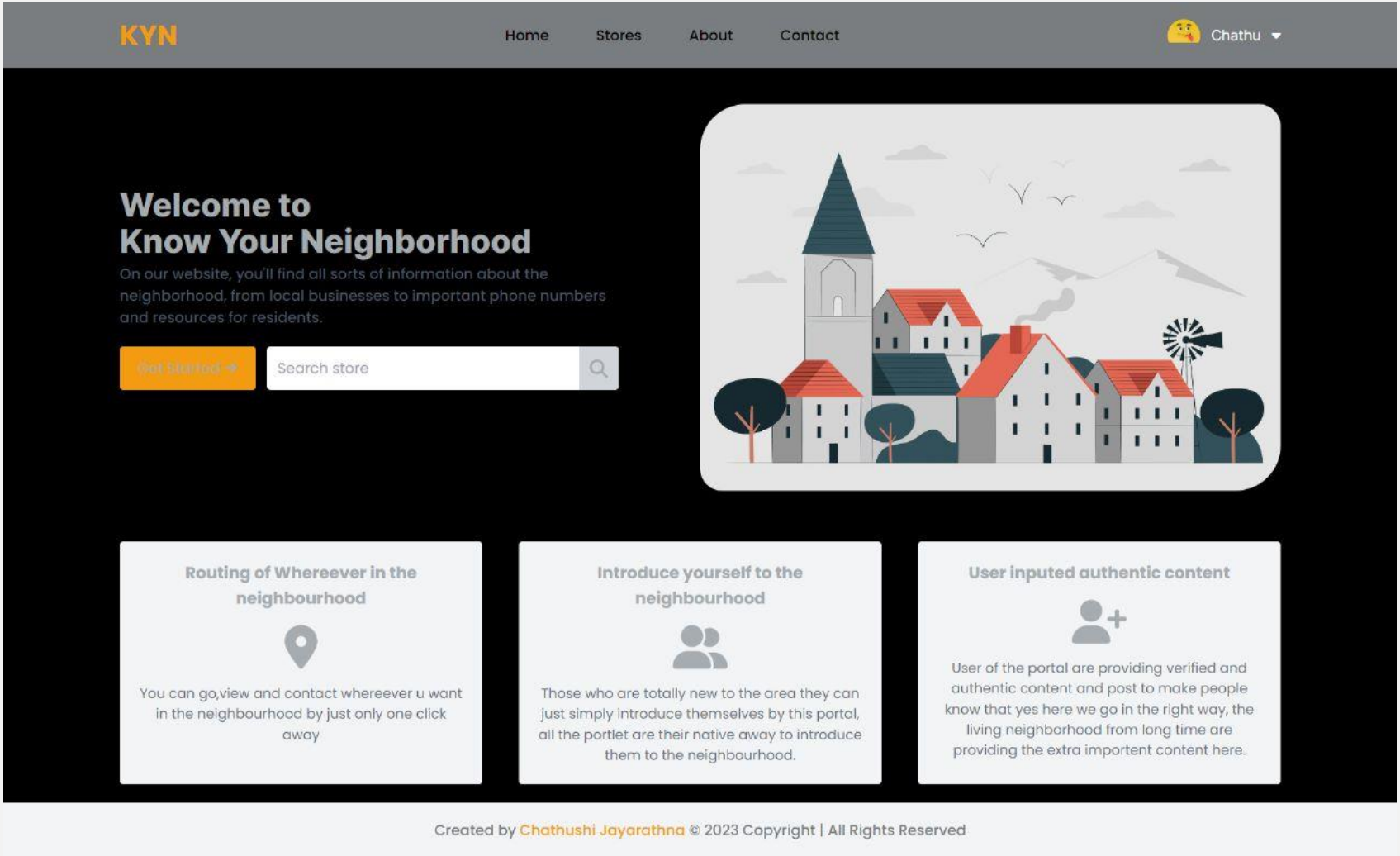
LITHAN

- **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.

6. The Project Demo

LITHAN

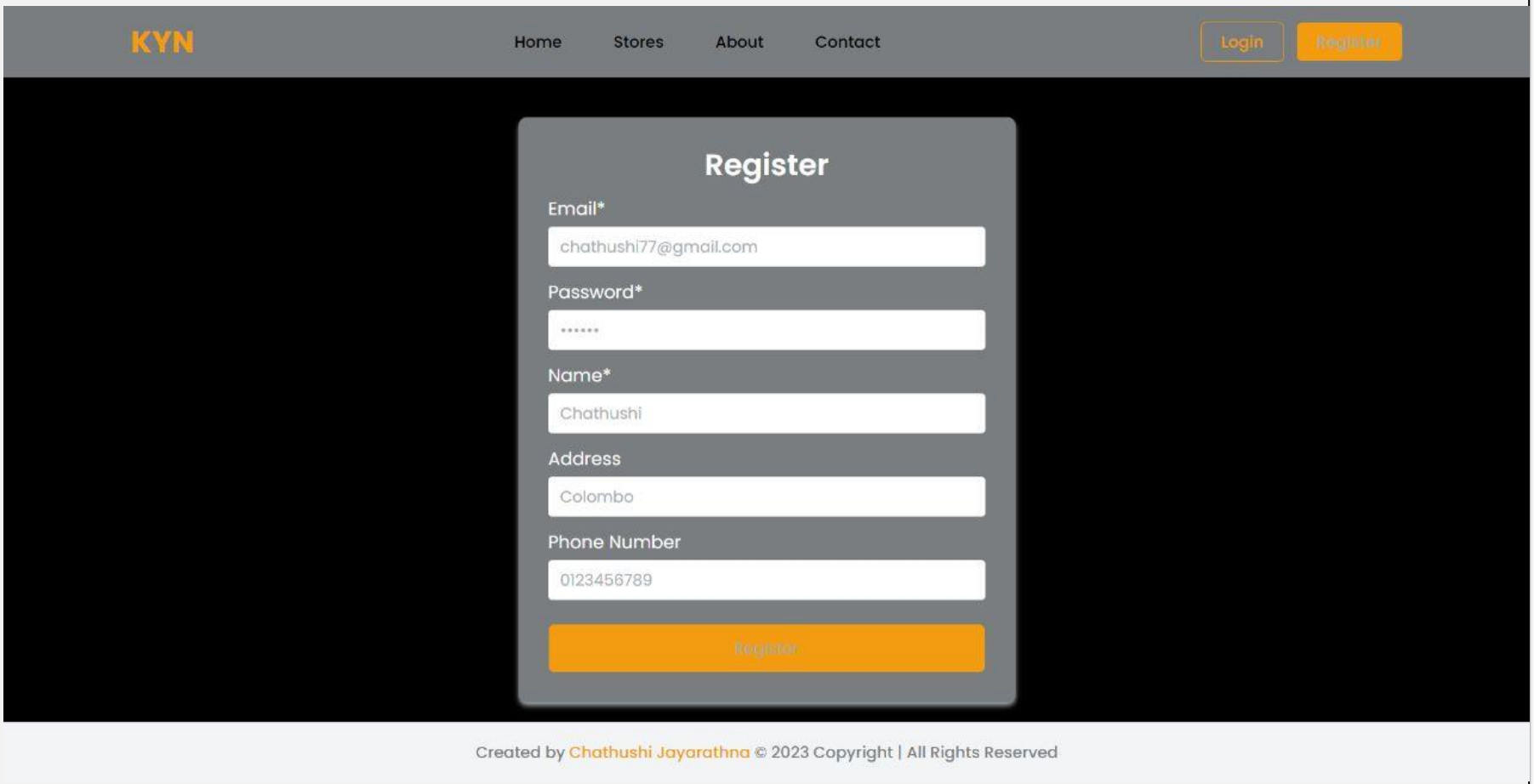
Home Page



6. The Project Demo

LITHAN

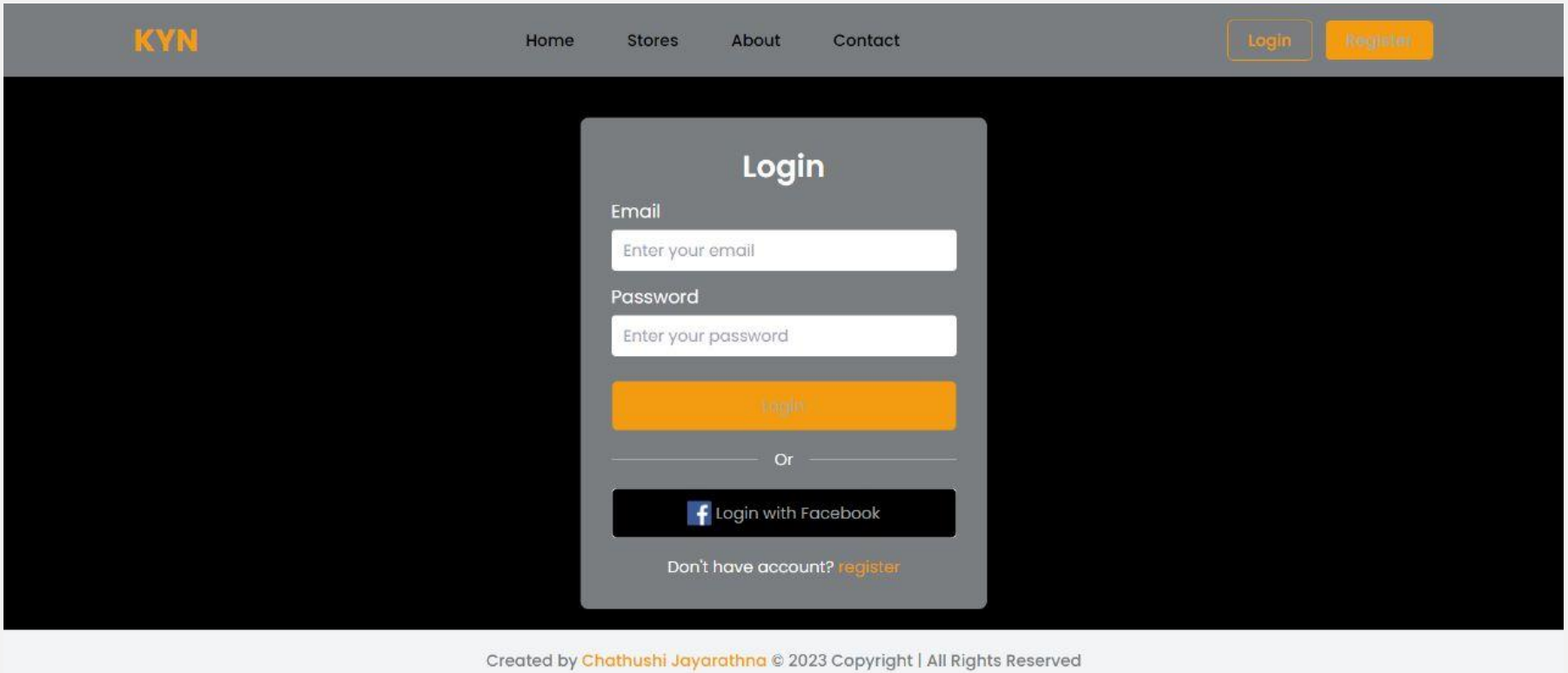
Registration



6. The Project Demo

LITHAN

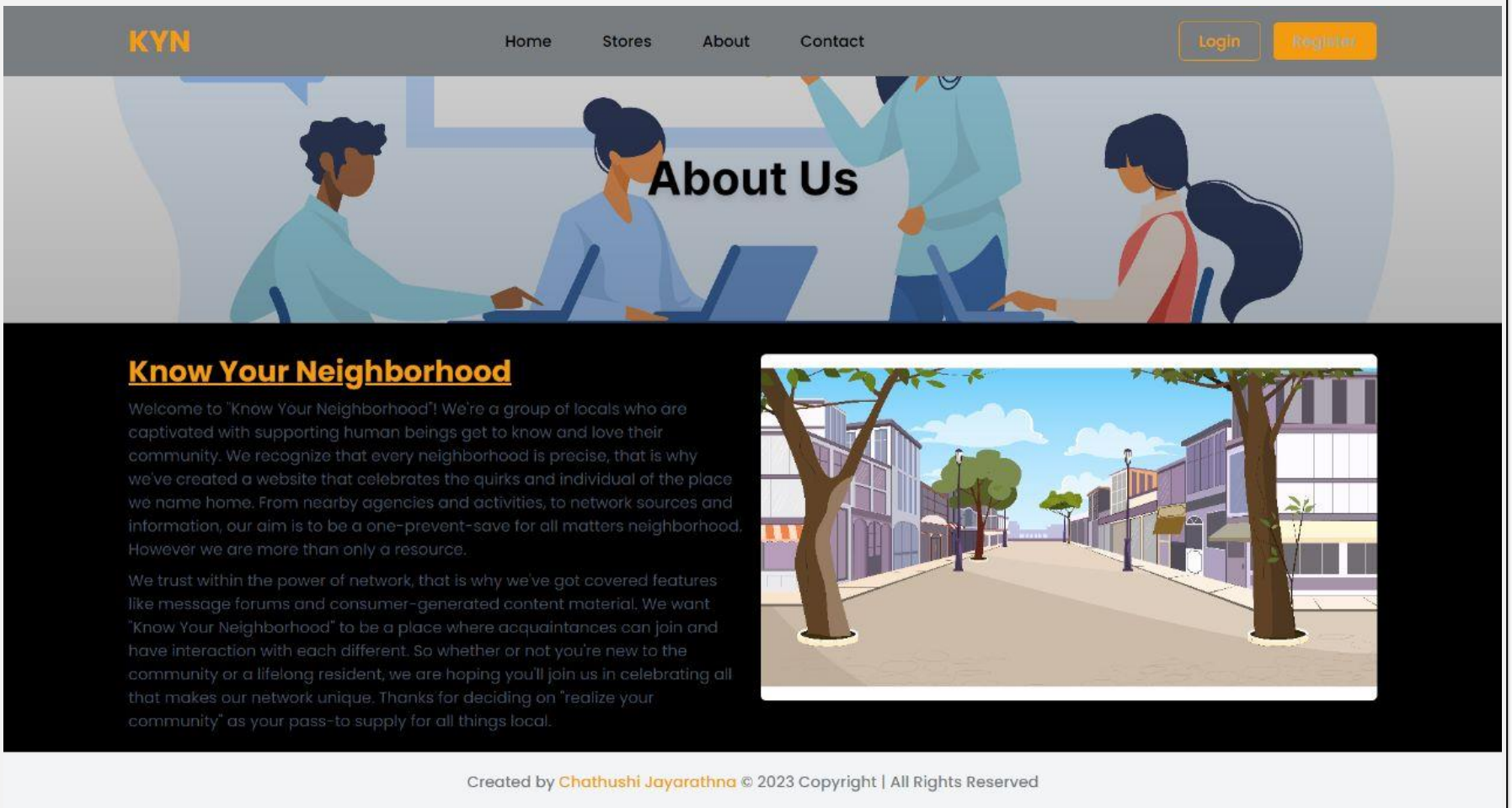
Login Page



6. The Project Demo

LITHAN

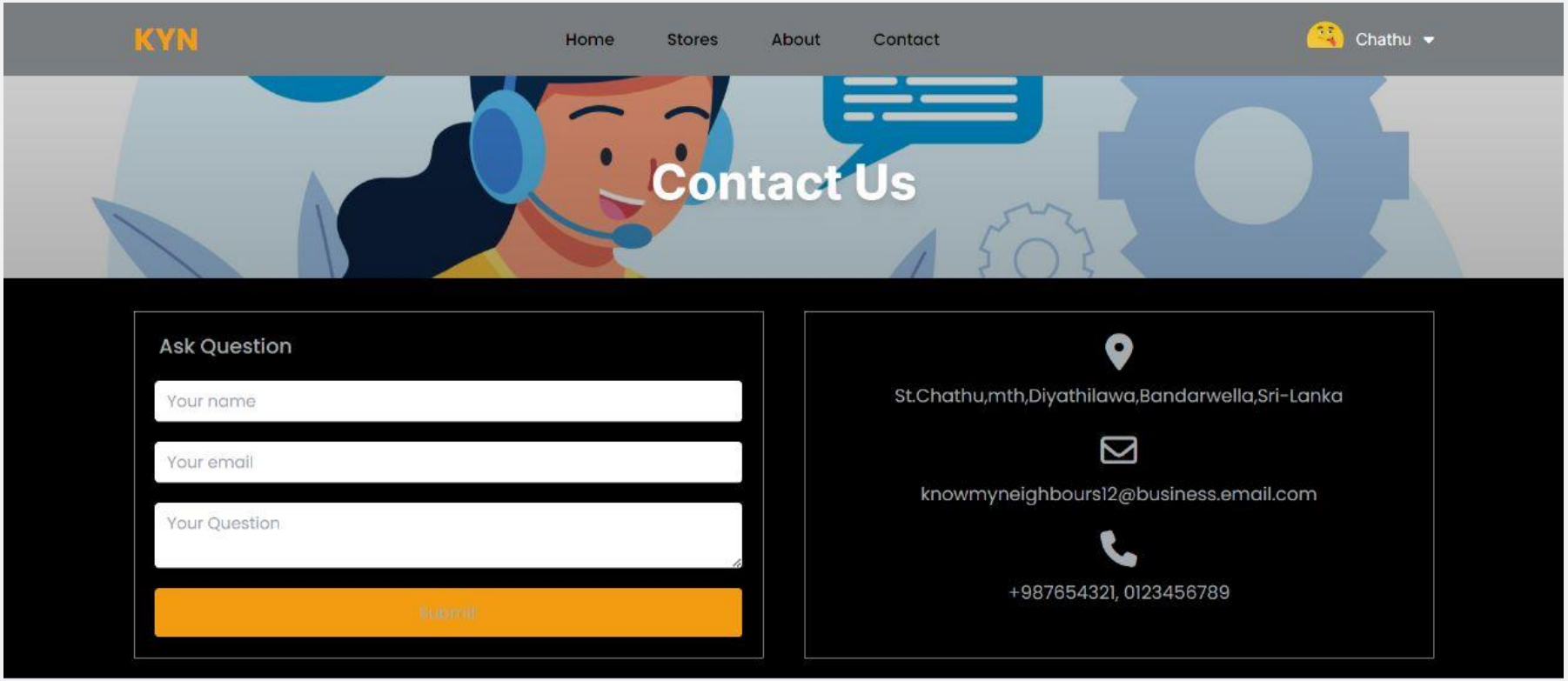
About Us



6. The Project Demo

LITHAN

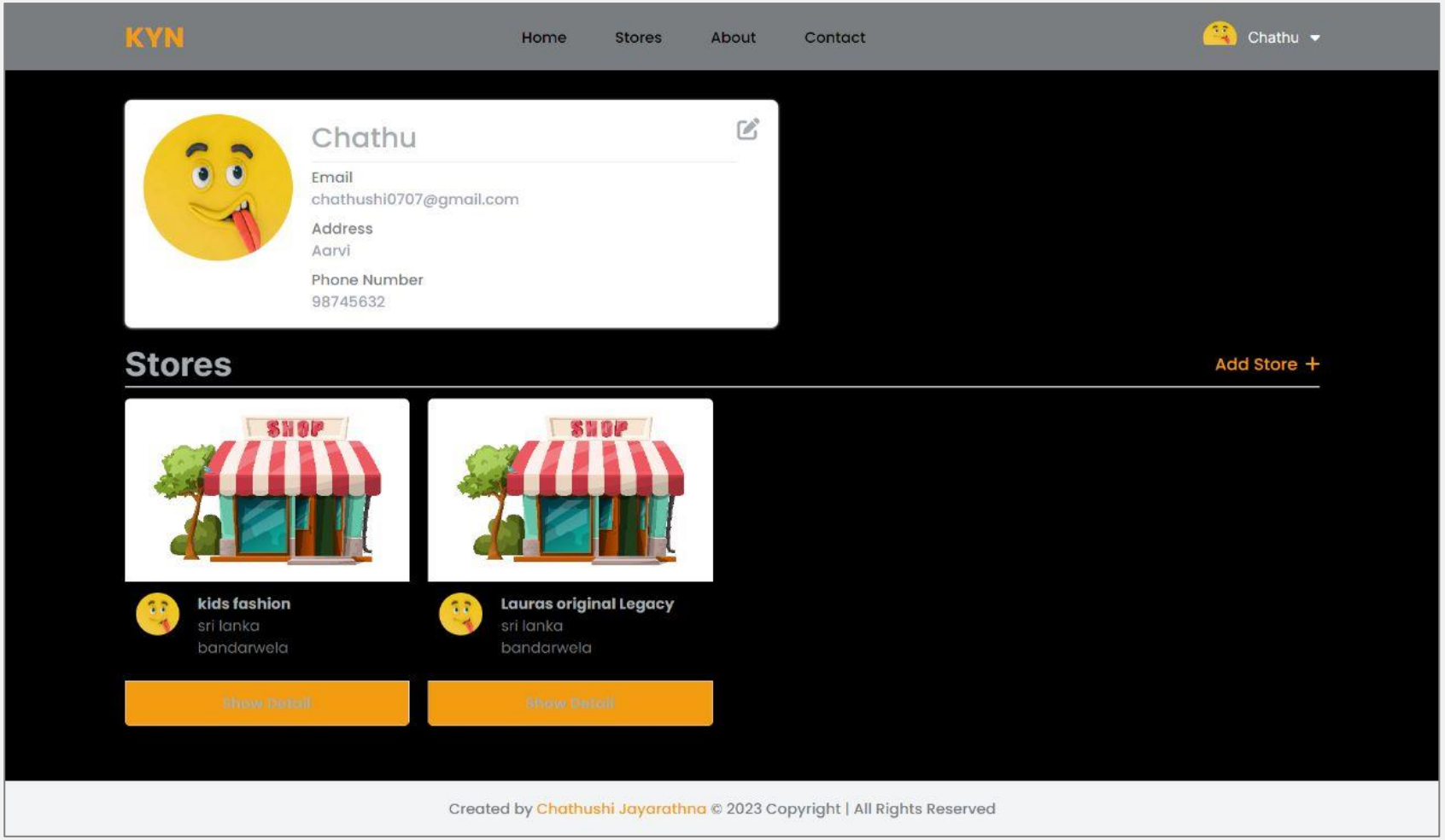
Contact Us Page



6. The Project Demo

LITHAN

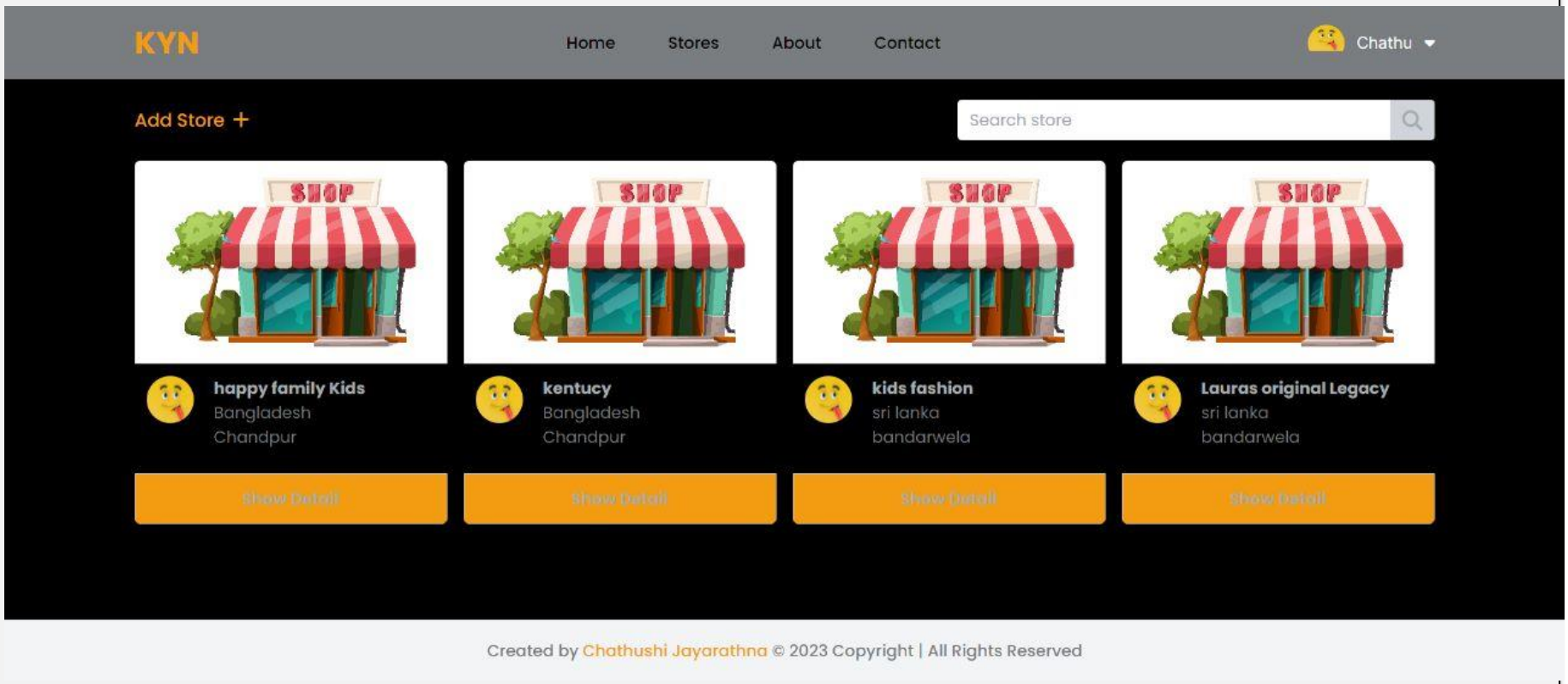
Profile Page



6. The Project Demo

LITHAN

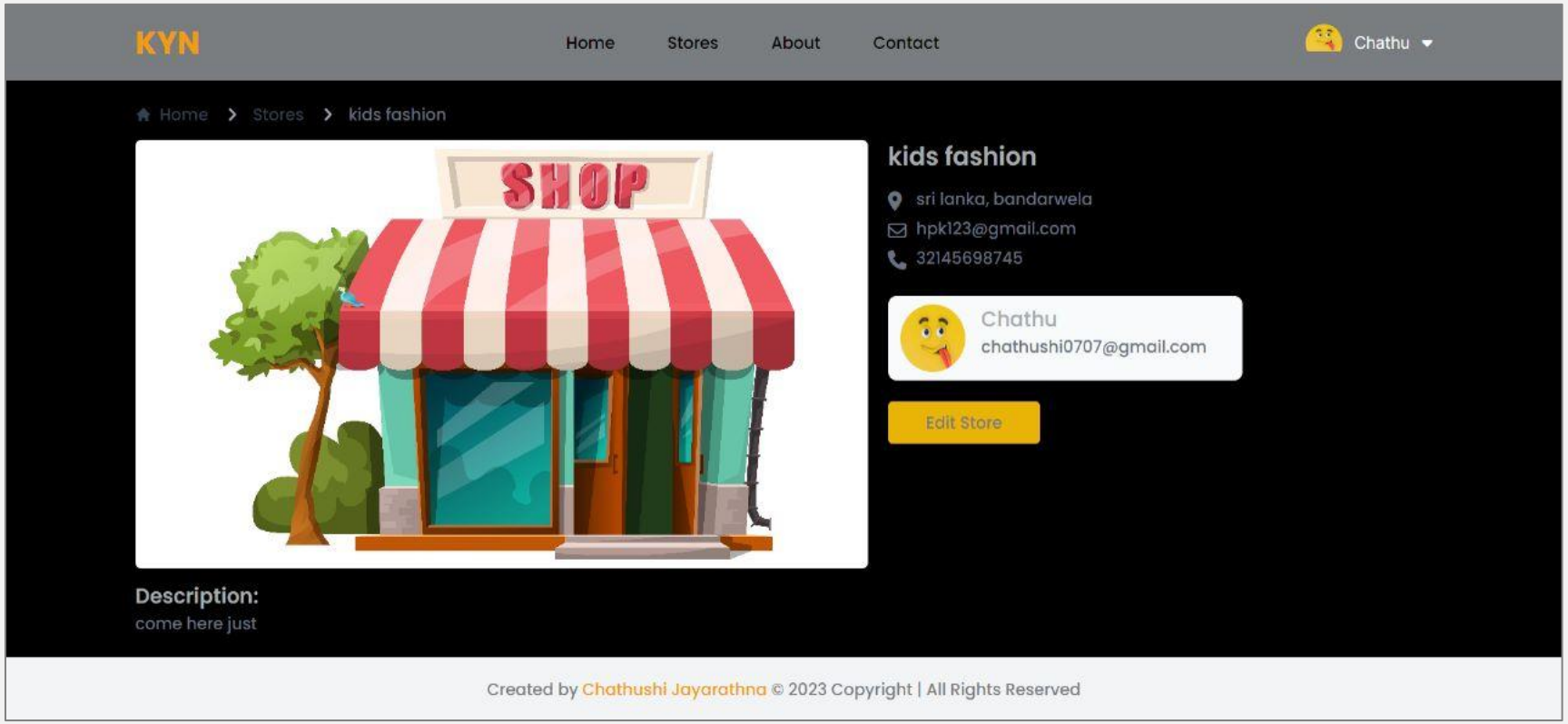
Stores Page



6. The Project Demo

LITHAN

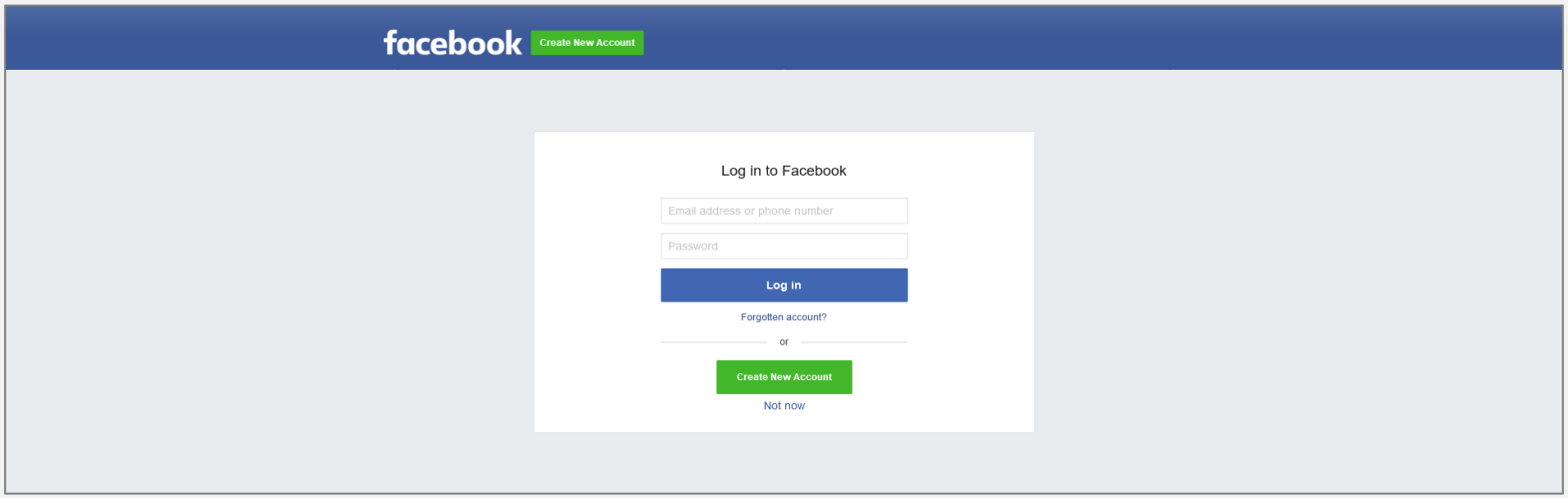
Store Details Page



6. The Project Demo

LITHAN

Facebook Page



7. The Strengths of Project API

LITHAN

- 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.

8. The Weaknesses of Project API

LITHAN

- 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.

9. Security Report

LITHAN

❑ **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.

❑ **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.

9. Security Report

LITHAN

❑ **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.