|  |  |
| --- | --- |
| 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) |

**Project Report**

|  |  |  |  |
| --- | --- | --- | --- |
| Student name | | Assessor name | |
| Chathushi Jayarathna | | MS.Aravinder Kaur | |
| Date issued | Completion date | | Submitted on |
| 2 January 2023 | 11 January 2023 | | 30 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: **Chathushi**  Date: 30 January 2023 |

Table of Contents

[**1)** **Project Background** 4](#_Toc126002333)

[**2)** **Project Objectives** 5](#_Toc126002334)

[2.1 Project Requirement Specification 5](#_Toc126002335)

[2.2 Tools & platforms used 6](#_Toc126002336)

[3) Explain what API is, its role and need for API and research existing APIs. 10](#_Toc126002337)

[3.1 What is API? 10](#_Toc126002338)

[3.2 API role and need for API 10](#_Toc126002339)

[3.3 Research existing APIs 11](#_Toc126002340)

[4) Relationship between API and SDK. 11](#_Toc126002341)

[5) Types of API and its uses. 13](#_Toc126002342)

[6) Identify the potential security issues with API and critically evaluate the suitable API for a given scenario or your selected application. 14](#_Toc126002343)

[7) Analyze the given scenario, identify the requirements and select the appropriate API. 16](#_Toc126002344)

[**7.1** **Requirements in the “Know-Your-Neighborhood” project.** 16](#_Toc126002345)

[8) Develop relevant wireframes for using the API for specific purposes. 17](#_Toc126002346)

[**9)** **Identify your scope and target platform.** 17](#_Toc126002347)

[10) Evaluate and justify your choice of APIs for your application. (Shows security for the selected API.) 18](#_Toc126002348)

[10.1 Security of selected API for the application 19](#_Toc126002349)

[11) Different types of the backend, frontend, and API implementation process 20](#_Toc126002350)

[12) Discuss a range of suitable development environments for front-end and back-end to develop an application 22](#_Toc126002351)

[13) Develop a backend and Web service using selected development environment for given scenario 23](#_Toc126002352)

[15) Construct the application which implements the selected API in Task 2. 34](#_Toc126002353)

[16) Implement white Box testing for the developed API of your Application 38](#_Toc126002354)

[16.1 What is white box testing? 38](#_Toc126002355)

[16.2 Why we use white box testing? 38](#_Toc126002356)

[17) Conduct Black Box testing (UAT testing) of your developed application and show the evidence for each test case. 40](#_Toc126002357)

[**17.1** **What is black box testing?** 40](#_Toc126002358)

[17.2 Benefits of black box testing 41](#_Toc126002359)

[18) Review your developed API, and identify the strength and weaknesses of API. 48](#_Toc126002360)

[19) Provide data security report of your developed application. 49](#_Toc126002361)

[**20)** **Developed Pages** 51](#_Toc126002362)

Table of Figures

[Figure 1:Screenshot of Spring Boot 8](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003216)

[Figure 2:Screenshot of MySQL Workbench 8](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003217)

[Figure 3:Screenshot of Spring Initializer 9](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003218)

[Figure 4:Screenshot of VS Code 9](#_Toc126003219)

[Figure 5:Screenshot of Diagram.net 10](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003220)

[Figure 6:Screenshot of Postman 10](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003221)

[Figure 7:Screenshot of MS Word 11](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003222)

[Figure 8:Screenshot of MS PowerPoint 11](#_Toc126003223)

[Figure 9:Wireframe of Facebook Login 19](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003224)

[Figure 10:Server Side Authentication 21](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003225)

[Figure 11:Screenshot of UserPrinciple.java 25](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003226)

[Figure 12:Screenshot of UserDetailsServiceImpl.java 26](#_Toc126003227)

[Figure 13:Screenshot of AuthController.java 28](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003228)

[Figure 14:Screenshot of FacebookOAuth2UserInfo.java 29](#_Toc126003229)

[Figure 15:Screenshot of HttpCookieOAuth2AuthorizationRequestRepository.java 30](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003230)

[Figure 16:Screenshot of Auth-context.js 32](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003231)

[Figure 17:Screenshot of FacebookLogin.jsx 33](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003232)

[Figure 18:Screenshot of LoginPage.jsx 35](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003233)

[Figure 19: Application Construction using Facebook Login API step 1 36](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003234)

[Figure 20:Application Construction using Facebook Login API step 2 37](#_Toc126003235)

[Figure 21:Application Construction using Facebook Login API step 3 37](#_Toc126003236)

[Figure 22:Application Construction using Facebook Login API step 4 38](#_Toc126003237)

[Figure 23:Application Construction using Facebook Login API step 5 38](#_Toc126003238)

[Figure 24: Application Construction using Facebook Login API (Application.yml) 39](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003239)

[Figure 25: Postman Testing - Registration 41](#_Toc126003240)

[Figure 26:Postman Testing - Login 41](#_Toc126003241)

[Figure 27:Postman Testing - Get user Login 41](#_Toc126003242)

[Figure 28:Black box testing 42](#_Toc126003243)

[Figure 29:Screenshot of Home Page 53](#_Toc126003244)

[Figure 30:Screenshot of Login Page 53](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003245)

[Figure 31:Screenshot of Facebook login 54](#_Toc126003246)

[Figure 32:Screenshot of Registration 54](#_Toc126003247)

[Figure 33:Screenshot of Stores 55](#_Toc126003248)

[Figure 34:Screenshot of Screenshot Store Details 55](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003249)

[Figure 35:Screenshot of Add Stores 56](file:///E:\Module%2010%20(API)\BDSE04-API-Chathushi%20Jayarathna-Final%20Project%20Report.docx#_Toc126003250)

[Figure 36:Screenshot of Edit Store 56](#_Toc126003251)

[Figure 37:Screenshot of Profile Page 57](#_Toc126003252)

[Figure 38:Screenshot of Edit Profile Page 57](#_Toc126003253)

[Figure 39:Screenshot of About Us Page 58](#_Toc126003254)

[Figure 40:Screenshot of Contact Us Page 58](#_Toc126003255)

1. **Project Background**

The Know-Your-Neighborhood application developed using Spring Boot is enhanced by API access. Rebuilding the application using React.Js and Spring Boot as a grant, demonstrates using APIs to log in and log out. For that purpose, this application should set up a login button with an API. Here Spring Boot framework is used for the backend and react Js is used for the front end. Also the scope here is to test and compare different existing APIs sample wise, assess their suitability, identify potential security issues and use selected APIs on existing websites create a login.

1. **Project Objectives**

* 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 applications using React JS.
* Be able to identify existing APIs and their uses in the 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 a login button with available APIs

* 1. Project Requirement Specification

The application can use existing APIs to log in and retrieve basic information such as name and email from the API. Users can also register/login manually or via social login (Google / Facebook).

The Know Your Neighborhood website consists of the following main pages:

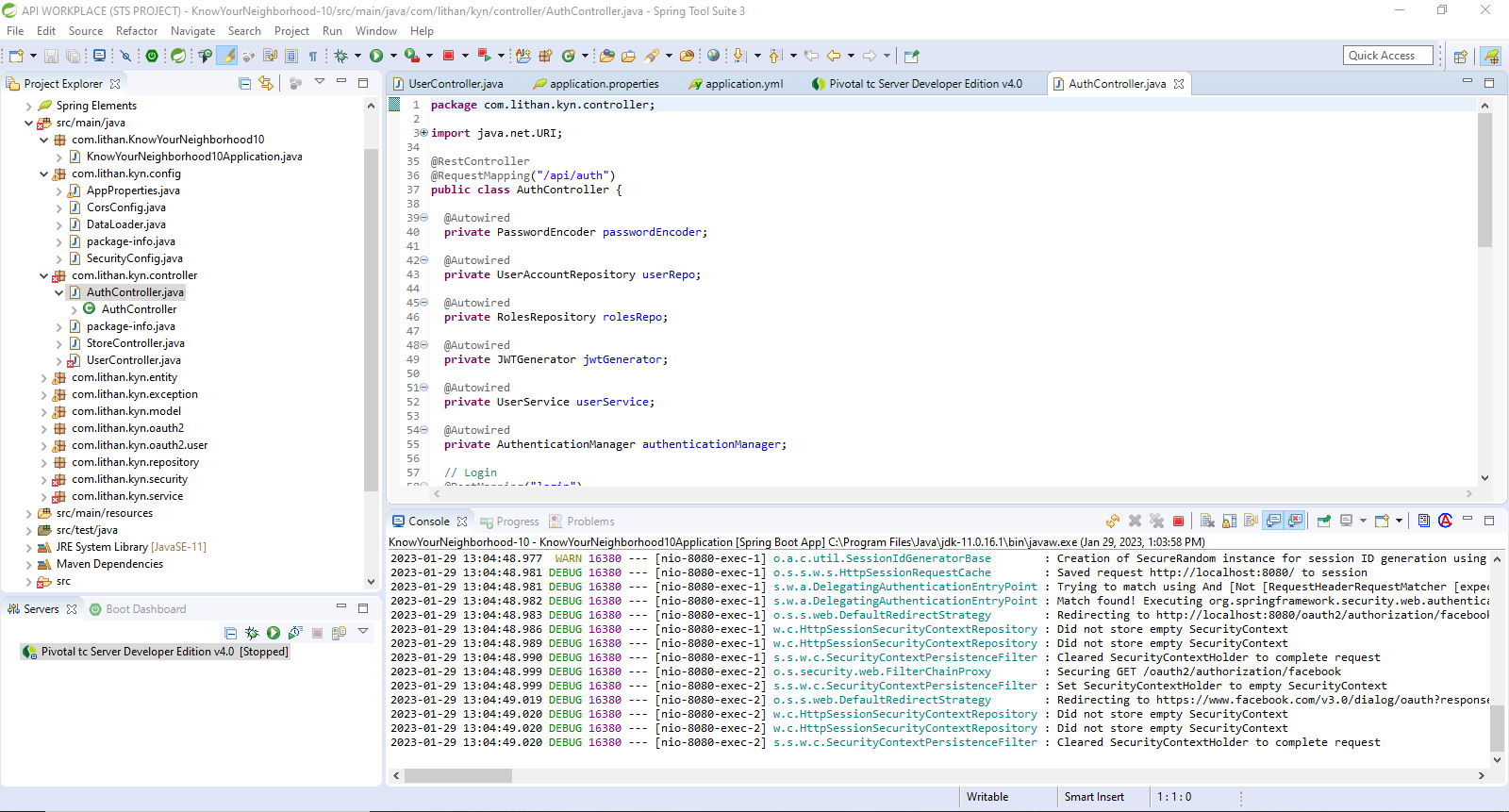
* Home Page
* Registration Page
* Login Pages with API link
* Contact us Page
* About us Page
* Dashboard
  1. Tools & platforms used
* Spring Boot - Backend

Figure :Screenshot of Spring Boot

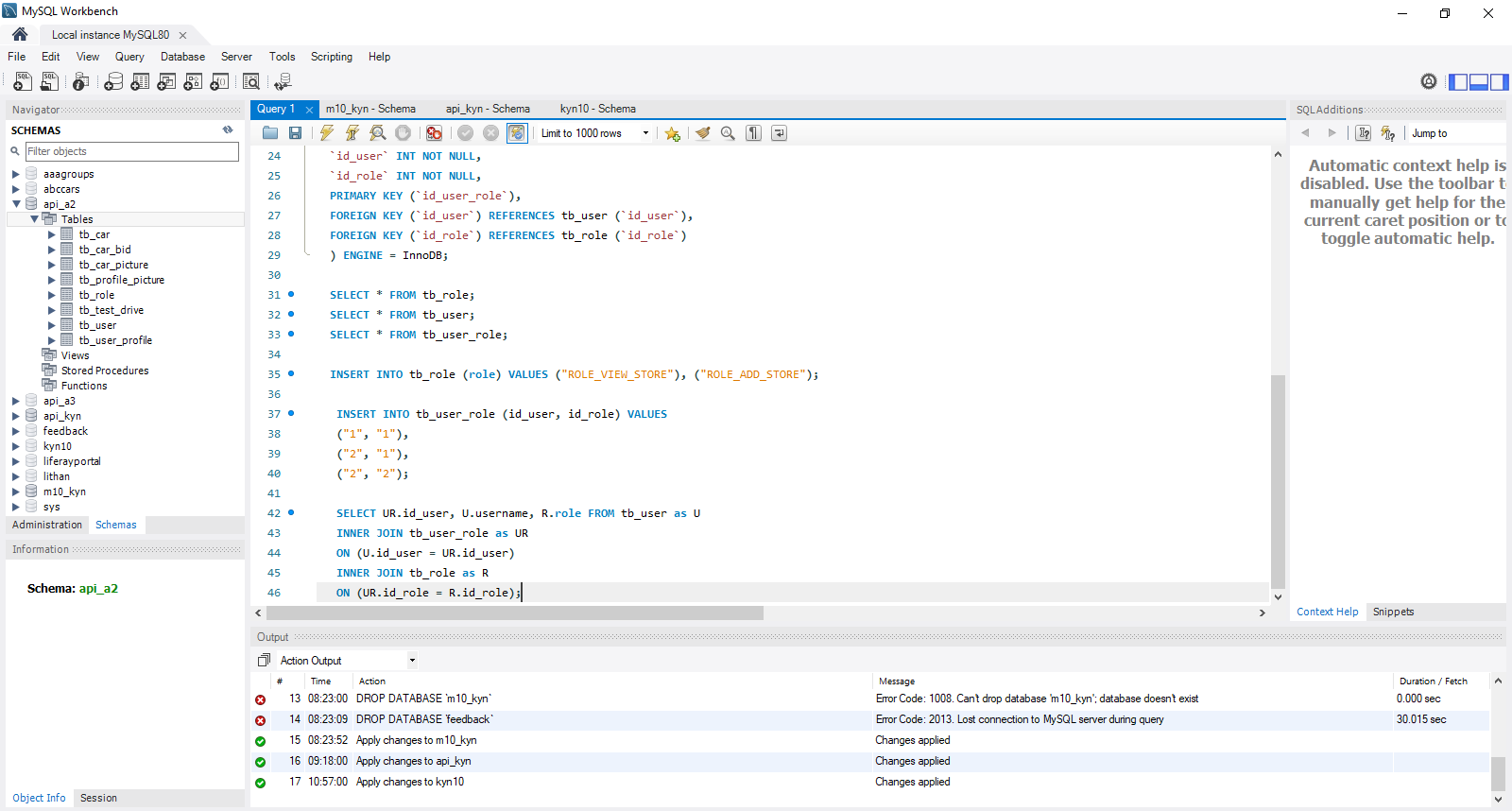
* MySQL Workbench - Database

Figure :Screenshot of MySQL Workbench

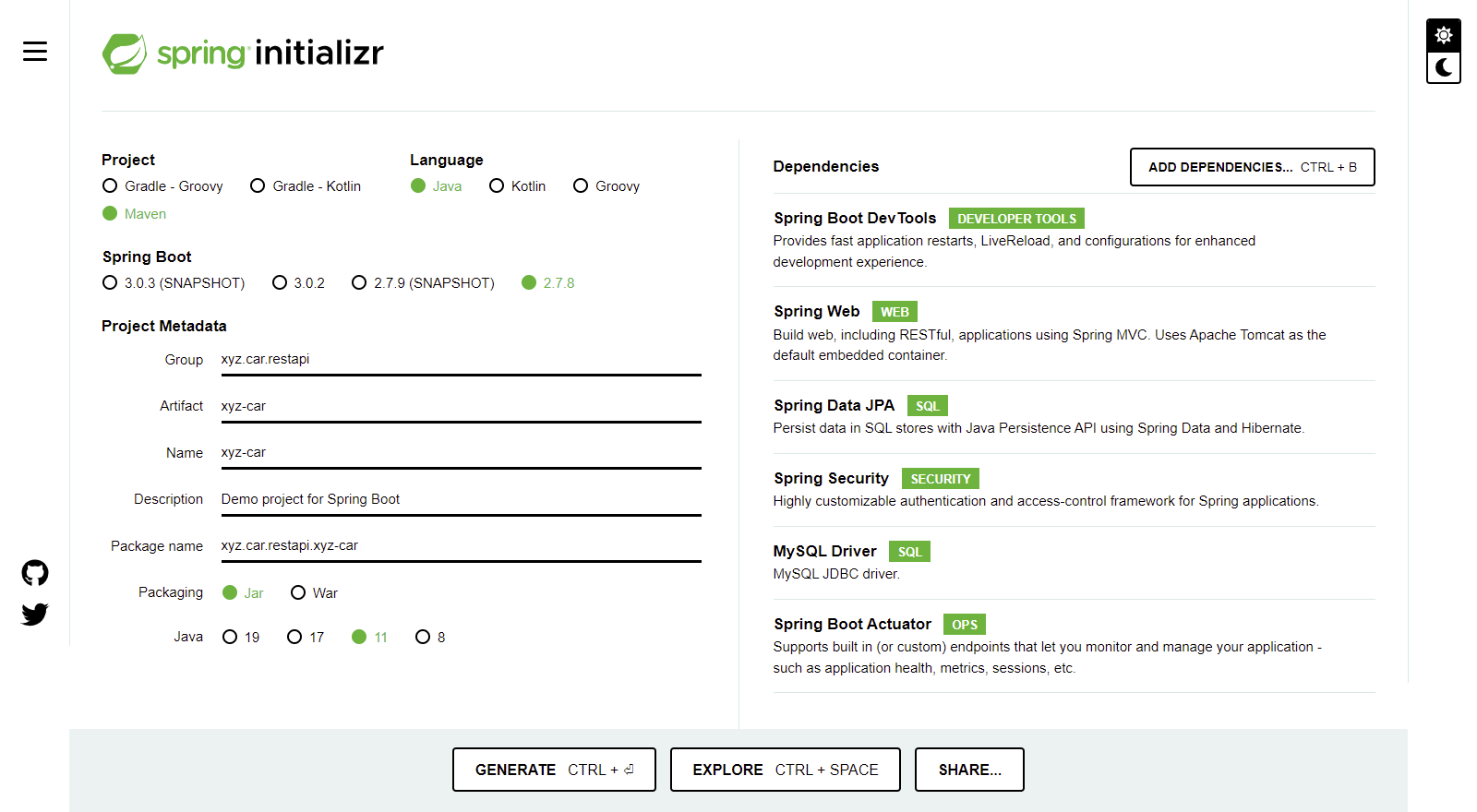
* Spring Initializer

Figure :Screenshot of Spring Initializer

* Visual Studio Code - Frontend

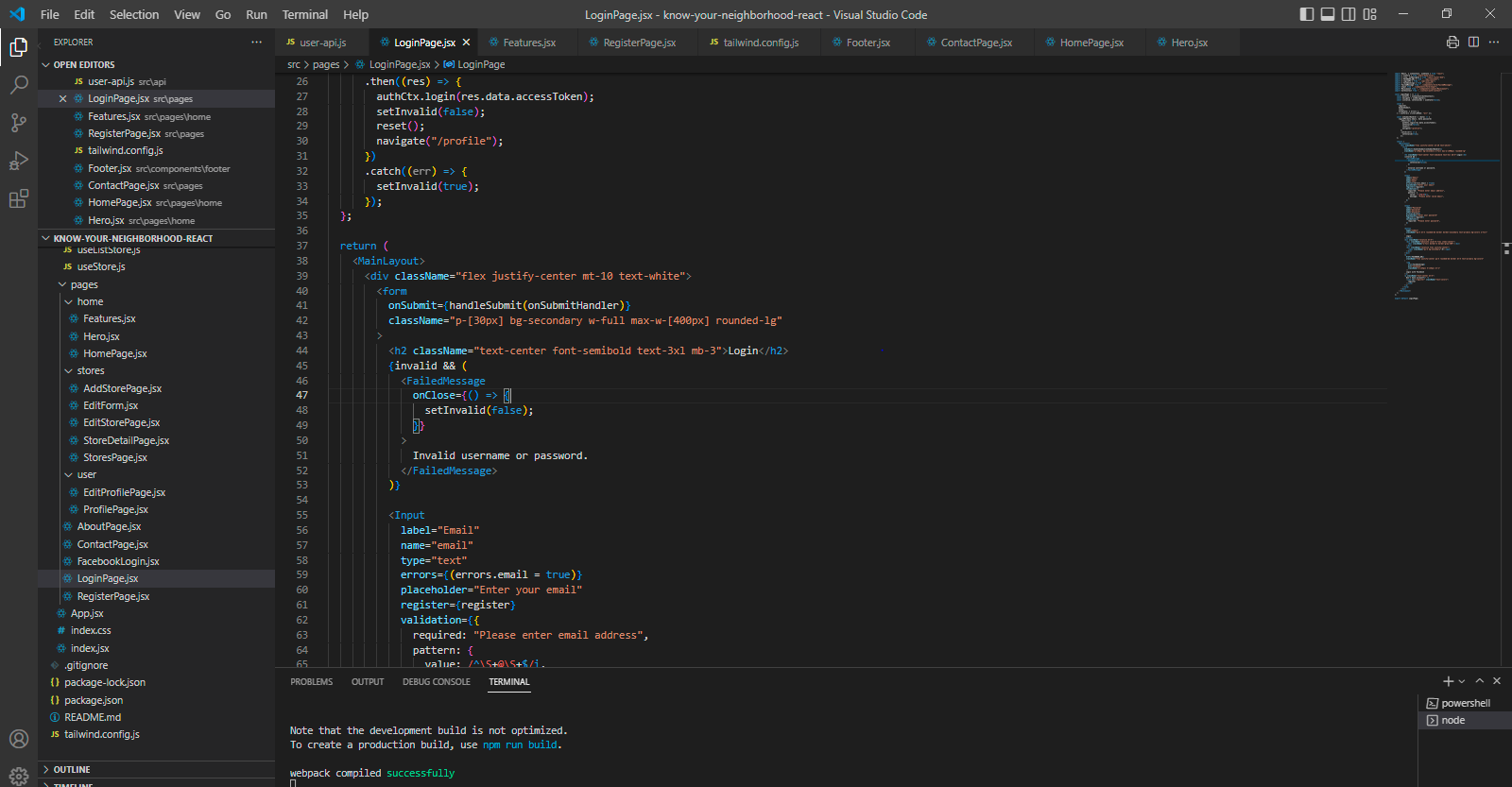


Figure :Screenshot of VS Code

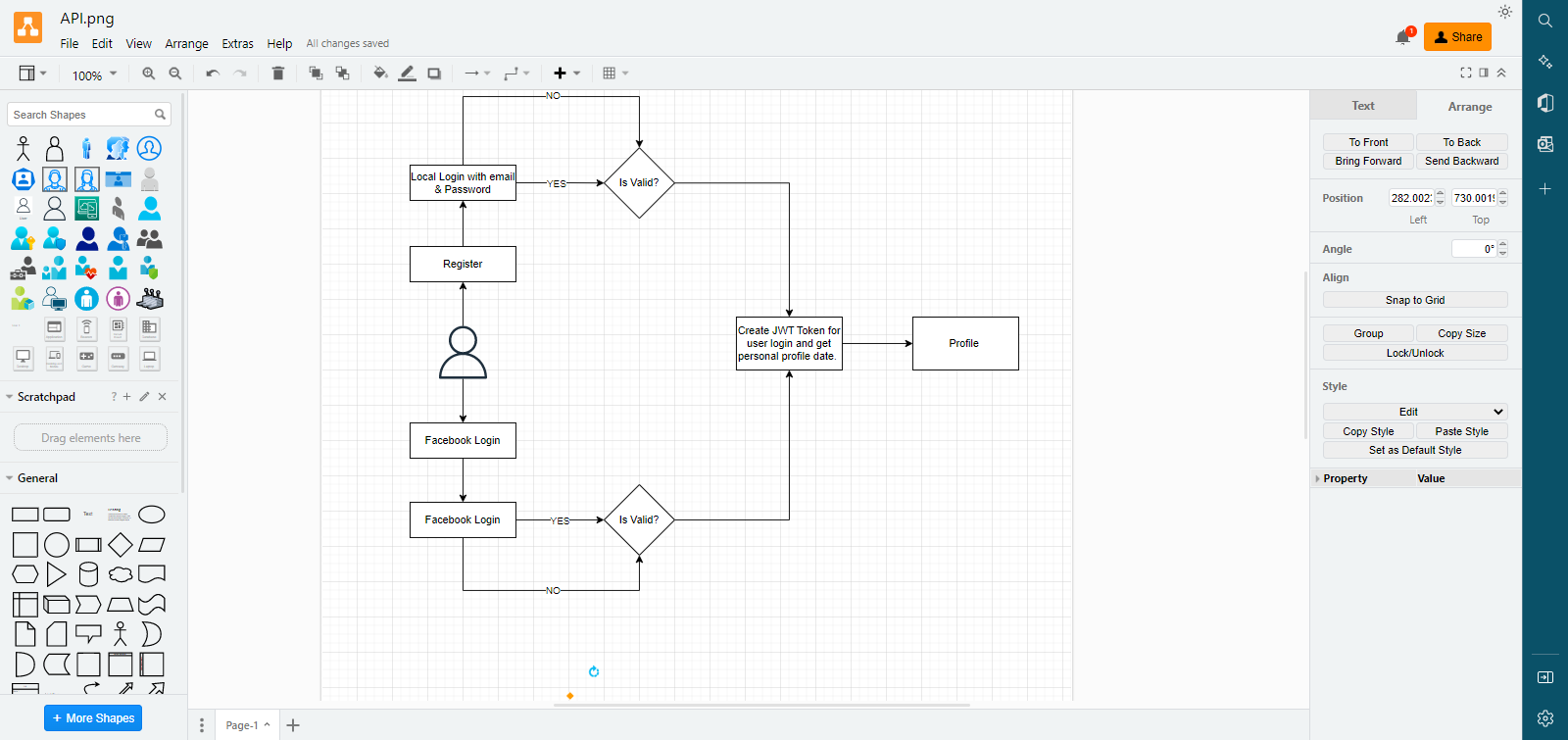
* Diagram.Net

Figure :Screenshot of Diagram.net

* Postman

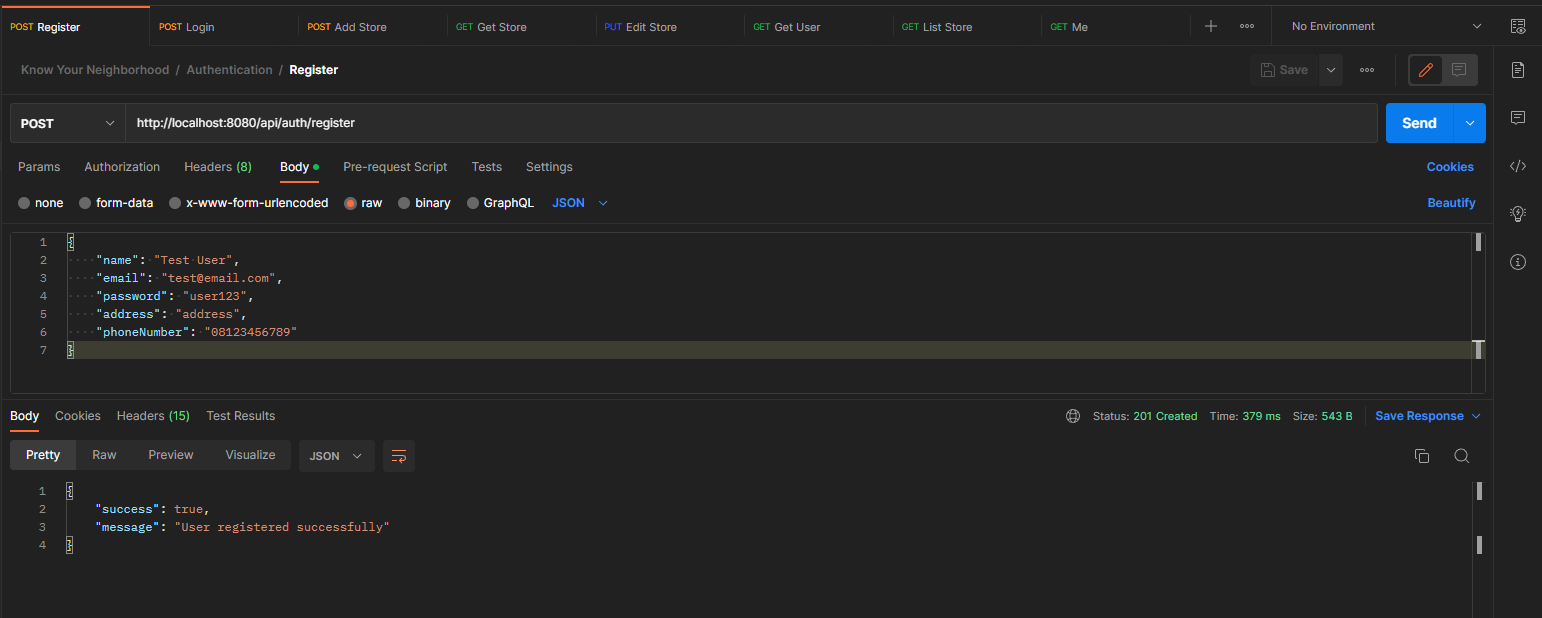
****

Figure :Screenshot of Postman

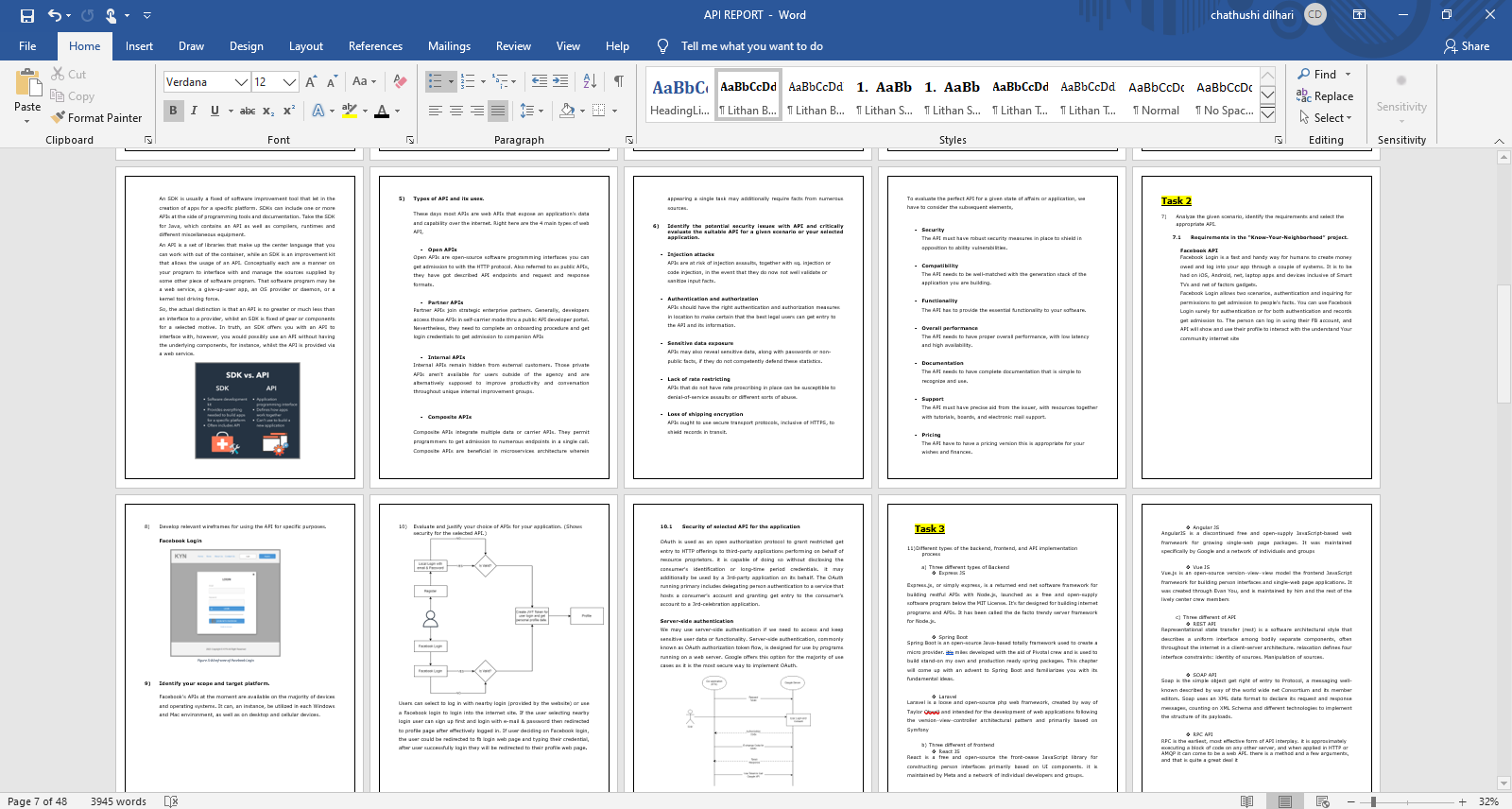
* **Microsoft Word**

Figure :Screenshot of MS Word

* **Microsoft PowerPoint**

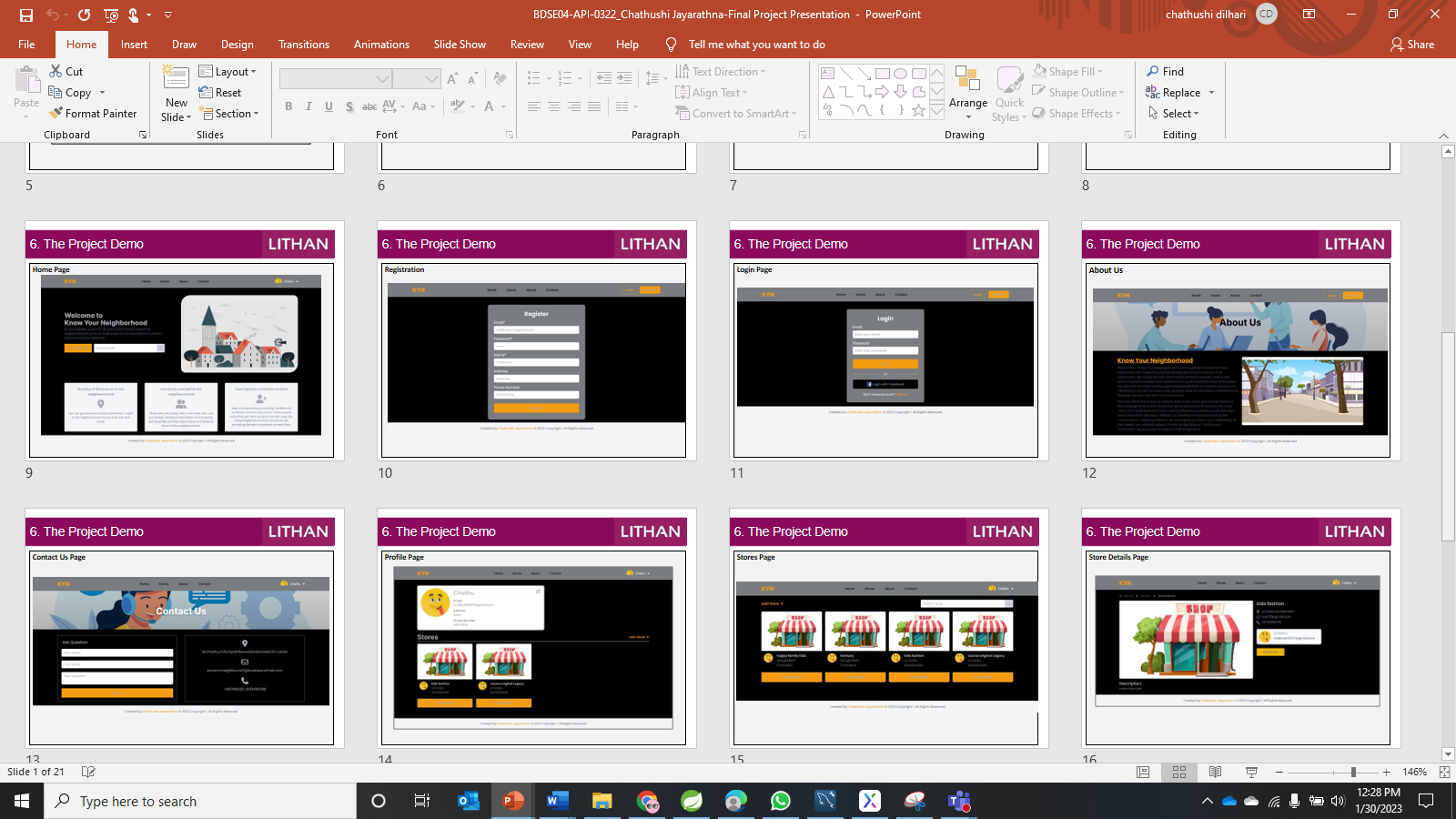


Figure :Screenshot of MS PowerPoint

Task 1

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

An application programming interface (API) is code that permits two software program packages to communicate. An API defines how a developer should request offerings from an operating machine (OS) or different utility, and expose data within one-of-a-kind contexts and throughout more than one channels.

Any statistics can be shared with an utility programming interface. APIs are implemented via function calls composed of verbs and nouns; the specified syntax is described in the documentation of the software being called. for example, on a actual estate internet site, one API would possibly put up available actual estate houses via geography, while a 2nd API provides current hobby quotes and a third offers a mortgage calculator.

* 1. API role and need for API

APIs are used in a huge variety of contexts, such as operating systems, application software programs, and web-based systems. They may be used to permit distinct software program packages to communicate with every different, to permit a front-end web application to access information from a back-end database or to allow a cell app to connect to a community-based service.

The want for APIs arises because extraordinary software systems regularly want to be incorporated with each other to be able to change statistics or provide positive services. as an example, a cellular app that enables users to discover nearby restaurants may use an API to fetch data from a database of restaurant listings. without an API, the mobile app would must immediately get entry to the database, which would be difficult to put into effect and maintain.

* 1. Research existing APIs

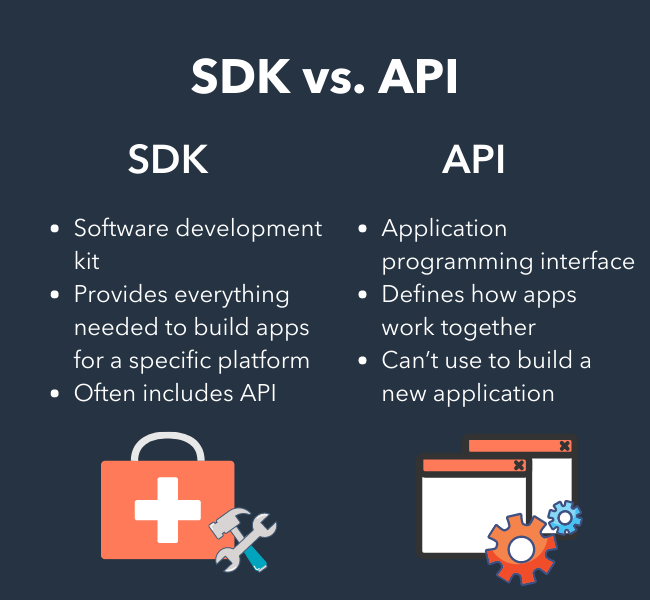
There are many exclusive APIs available for an extensive range of functions. A few APIs are open and to be had to anyone, at the same time as others are proprietary and only available to unique businesses or people. To analyze current APIs, you could seek online directories inclusive of programmable web or Rapid API, which list thousands of APIs in various classes. You may also look for APIs particular to a specific carrier or generation, including the Google Maps API or the Facebook API.

1. Relationship between API and SDK.

Two terms in software development that are frequently confused are API and SDK. API stands for “application Programming Interface” and refers to programming commands and standards for accessing a web tool or database. For example, a software program enterprise will often release its API publicly or privately to different software builders in order to design products powered by its service. An API may be packaged in an SDK, or “software improvement package “.

An SDK is usually a fixed of software improvement tool that let in the creation of apps for a specific platform. SDKs can include one or more APIs at the side of programming tools and documentation. Take the SDK for Java, which contains an API as well as compilers, runtimes and different miscellaneous equipment.

An API is a set of libraries that make up the center language that you can work with out of the container, while an SDK is an improvement kit that allows the usage of an API. Conceptually each are a manner on your program to interface with and manage the sources supplied by some other piece of software program. That software program may be a web service, a give-up-user app, an OS provider or daemon, or a kernel tool driving force.

So, the actual distinction is that an API is no greater or much less than an interface to a provider, whilst an SDK is fixed of gear or components for a selected motive. In truth, an SDK offers you with an API to interface with, however, you would possibly use an API without having the underlying components, for instance, whilst the API is provided via a web service.

1. Types of API and its uses.

These days most APIs are web APIs that expose an application's data and capability over the internet. Right here are the 4 main types of web API,

* Open APIs

Open APIs are open-source software programming interfaces you can get admission to with the HTTP protocol. Also referred to as public APIs, they have got described API endpoints and request and response formats.

* Partner APIs

Partner APIs join strategic enterprise partners. Generally, developers access those APIs in self-carrier mode thru a public API developer portal. Nevertheless, they need to complete an onboarding procedure and get login credentials to get admission to companion APIs

* Internal APIs

Internal APIs remain hidden from external customers. Those private APIs aren't available for users outside of the agency and are alternatively supposed to improve productivity and conversation throughout unique internal improvement groups.

* Composite APIs

Composite APIs integrate multiple data or carrier APIs. They permit programmers to get admission to numerous endpoints in a single call. Composite APIs are beneficial in microservices architecture wherein appearing a single task may additionally require facts from numerous sources.

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

* **Injection attacks**

APIs are at risk of injection assaults, together with sq. injection or code injection, in the event that they do now not well validate or sanitize input facts.

* **Authentication and authorization**

APIs should have the right authentication and authorization measures in location to make certain that the best legal users can get entry to the API and its information.

* **Sensitive data exposure**

APIs may also reveal sensitive data, along with passwords or non-public facts, if they do not competently defend these statistics.

* **Lack of rate restricting**

APIs that do not have rate proscribing in place can be susceptible to denial-of-service assaults or different sorts of abuse.

* **Loss of shipping encryption**

APIs ought to use secure transport protocols, inclusive of HTTPS, to shield records in transit.

To evaluate the perfect API for a given state of affairs or application, we have to consider the subsequent elements,

* **Security**

The API must have robust security measures in place to shield in opposition to ability vulnerabilities.

* **Compatibility**

The API needs to be well-matched with the generation stack of the application you are building.

* **Functionality**

The API has to provide the essential functionality to your software.

* **Overall performance**

The API needs to have proper overall performance, with low latency and high availability.

* **Documentation**

The API needs to have complete documentation that is simple to recognize and use.

* **Support**

The API must have precise aid from the issuer, with resources together with tutorials, boards, and electronic mail support.

* **Pricing**

The API have to have a pricing version this is appropriate for your wishes and finances.

**Task 2**

1. Analyze the given scenario, identify the requirements and select the appropriate API.
   1. **Requirements in the “Know-Your-Neighborhood” project.**

**Facebook API**

Facebook Login is a fast and handy way for humans to create money owed and log into your app through a couple of systems. It is to be had on iOS, Android, net, laptop apps and devices inclusive of Smart TVs and net of factors gadgets.

Facebook Login allows two scenarios, authentication and inquiring for permissions to get admission to people's facts. You can use Facebook Login surely for authentication or for both authentication and records get admission to. The person can log in using their FB account, and API will show and use their profile to interact with the understand Your community internet site

1. Develop relevant wireframes for using the API for specific purposes.

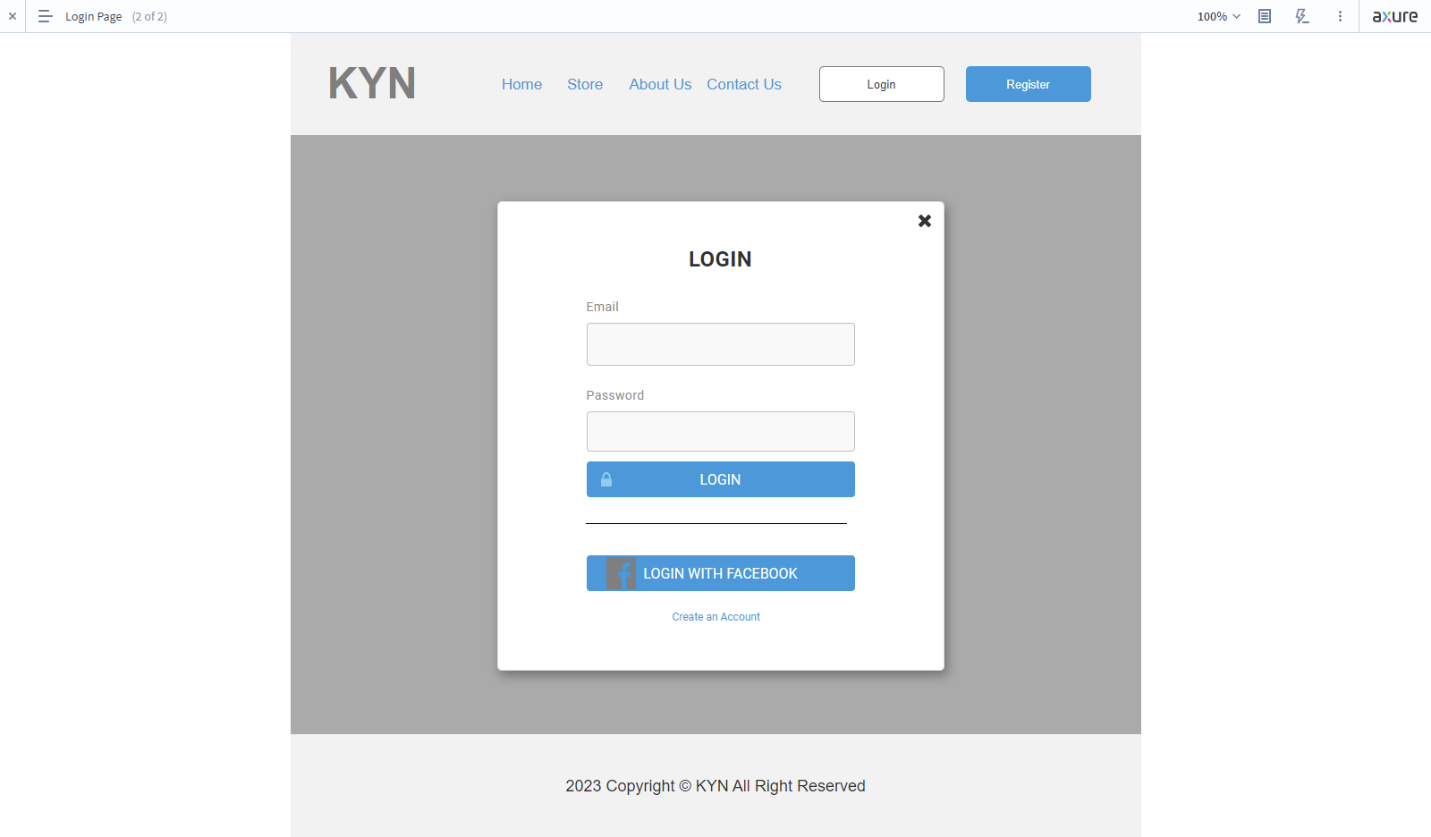
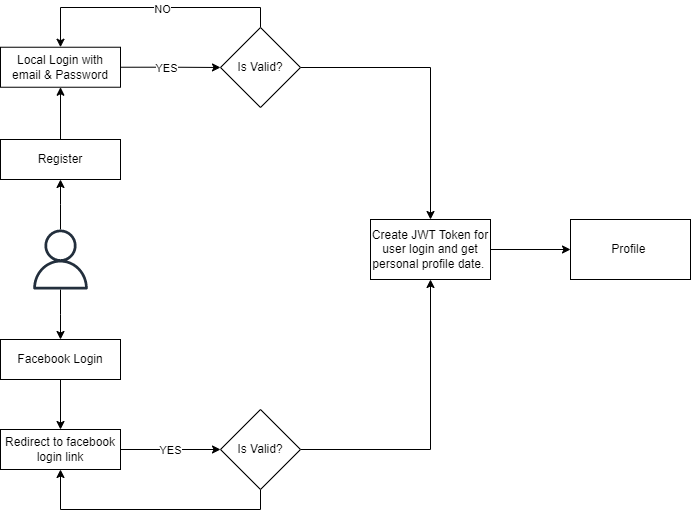
**Facebook Login**

Figure :Wireframe of Facebook Login

1. **Identify your scope and target platform.**

Facebook's APIs at the moment are available on the majority of devices and operating systems. It can, an instance, be utilized in each Windows and Mac environment, as well as on desktop and cellular devices.

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

Users can select to log in with nearby login (provided by the website) or use a Facebook login to login into the internet site. If the user selecting nearby login user can sign up first and login with e-mail & password then redirected to profile page after effectively logged in. If user deciding on Facebook login, the user could be redirected to fb login web page and typing their credential, after user successfully login they will be redirected to their profile web page**.**

* 1. Security of selected API for the application

OAuth is used as an open authorization protocol to grant restricted get entry to HTTP offerings to third-party applications performing on behalf of resource proprietors. it is capable of doing so without disclosing the consumer's identification or long-time period credentials. it may additionally be used by a 3rd-party application on its behalf. The OAuth running primary includes delegating person authentication to a service that hosts a consumer's account and granting get entry to the consumer's account to a 3rd-celebration application.

**Server-side authentication**

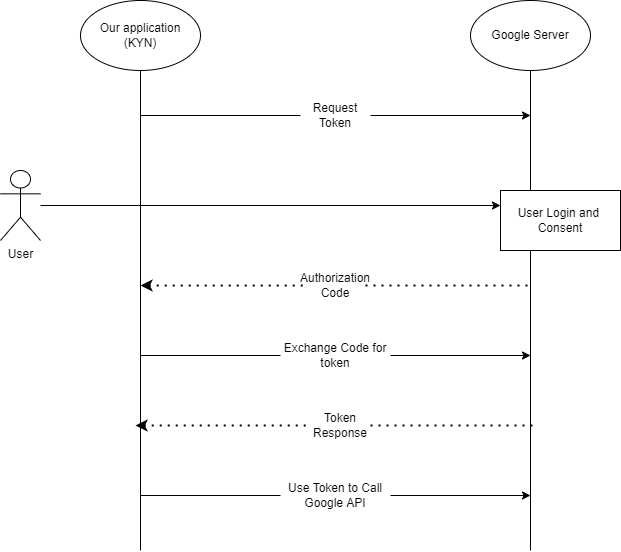
We may use server-side authentication if we need to access and keep sensitive user data or functionality. Server-side authentication, commonly known as OAuth authorization token flow, is designed for use by programs running on a web server. Google offers this option for the majority of use cases as it is the most secure way to implement OAuth.

Figure :Server Side Authentication

**Task 3**

1. Different types of the backend, frontend, and API implementation process
2. Three different types of Backend

* Express JS

Express.js, or simply express, is a returned end net software framework for building restful APIs with Node.js, launched as a free and open-supply software program below the MIT License. It's far designed for building internet programs and APIs. It has been called the de facto trendy server framework for Node.js.

* Spring Boot

Spring Boot is an open-source Java-based totally framework used to create a micro provider. it's miles developed with the aid of Pivotal crew and is used to build stand-on my own and production ready spring packages. This chapter will come up with an advent to Spring Boot and familiarizes you with its fundamental ideas.

* Laravel

Laravel is a loose and open-source php web framework, created by way of Taylor Otwell and intended for the development of web applications following the version–view–controller architectural pattern and primarily based on Symfony

1. Three different of frontend

* React JS

React is a free and open-source the front-cease JavaScript library for constructing person interfaces primarily based on UI components. it is maintained by Meta and a network of individual developers and groups.

* Angular JS

AngularJS is a discontinued free and open-supply JavaScript-based web framework for growing single-web page packages. It was maintained specifically by Google and a network of individuals and groups

* Vue JS

Vue.js is an open-source version–view–view model the frontend JavaScript framework for building person interfaces and single-web page applications. It was created through Evan You, and is maintained by him and the rest of the lively center crew members

1. Three different of API

* REST API

Representational state transfer (rest) is a software architectural style that describes a uniform interface among bodily separate components, often throughout the internet in a client-server architecture. relaxation defines four interface constraints: identity of sources. Manipulation of sources.

* SOAP API

Soap is the simple object get right of entry to Protocol, a messaging well-known described by way of the world wide net Consortium and its member editors. Soap uses an XML data format to declare its request and response messages, counting on XML Schema and different technologies to implement the structure of its payloads.

* RPC API

RPC is the earliest, most effective form of API interplay. it is approximately executing a block of code on any other server, and when applied in HTTP or AMQP it can come to be a web API. there is a method and a few arguments, and that is quite a great deal it

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

* **Visual Studio Code**

Visual Studio Code combines the simplicity of a source code editor with effective developer tooling, like intelligence code of completion and debugging. First and major, it's far an editor that receives from your way. The delightfully frictionless edit-construct-debug cycle approach less time fiddling with your surroundings, and extra time executing in your thoughts

* **WebStorm**

WebStorm is an incorporated development environment for JavaScript and related technologies. Like different JetBrains IDEs, it makes your development experience extra exciting, automating routine work and assisting you manage complex tasks effortlessly.

1. **Backend Development**

* **Visual Studio**

A general-purpose development environment that can be used for back-end development with a variety of languages, including C# and .NET.

* **Spring Tool Suite**

Spring tool Suite (STS) is a java IDE tailored for developing Spring-primarily based organization programs. it is less complicated, quicker, and extra convenient. And most importantly it's far based on Eclipse IDE. STS is unfastened, open-supply, and powered with the aid of VMware

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

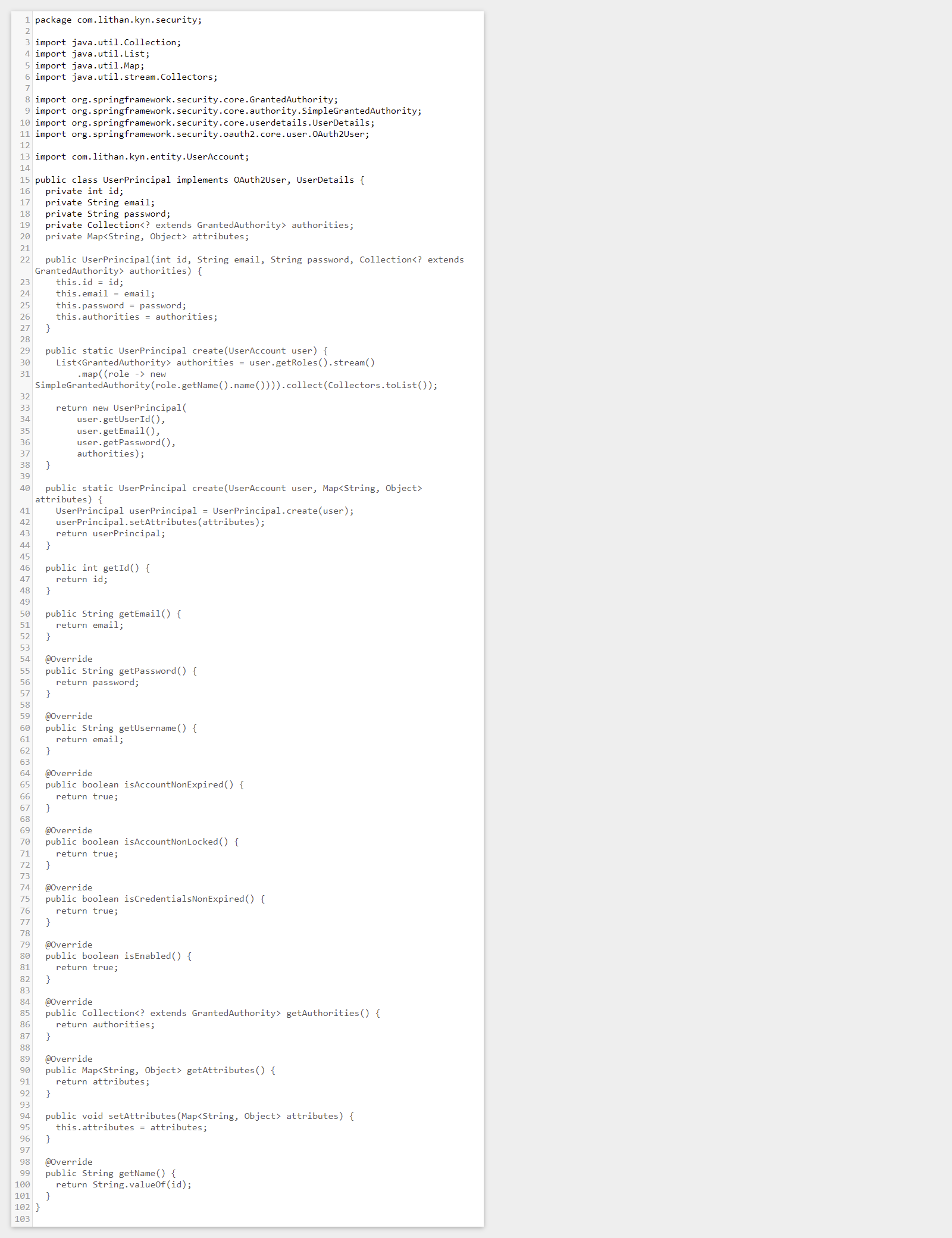
**UserPrinciple.java**

Figure :Screenshot of UserPrinciple.java

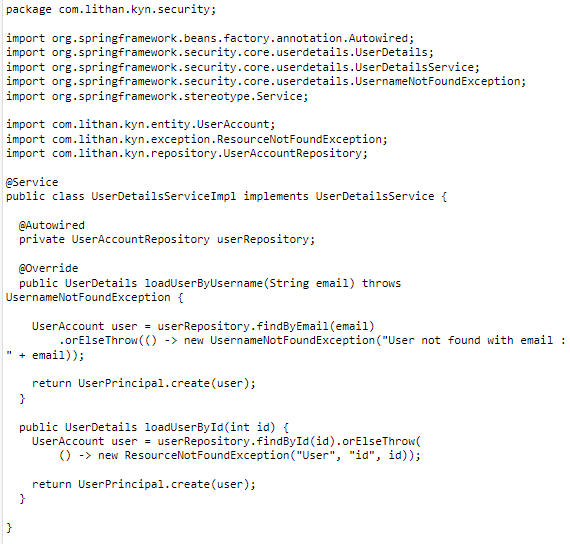
**UserDetailsServiceImpl.java**

Figure :Screenshot of UserDetailsServiceImpl.java

AuthController.java



Figure :Screenshot of AuthController.java

Backend Facebook Login

FacebookOAuth2UserInfo.java

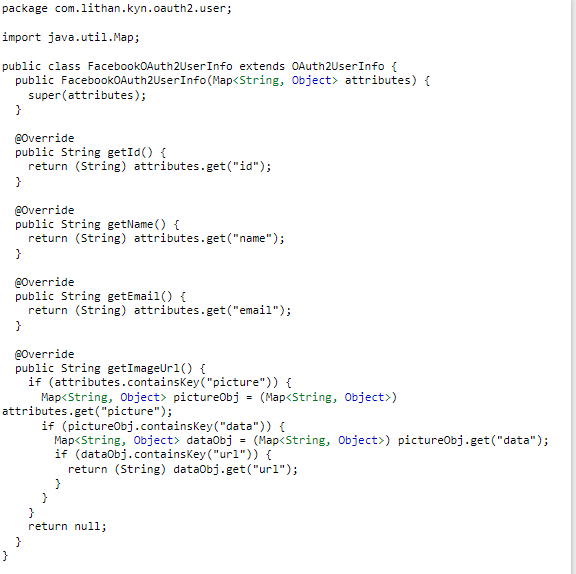


Figure :Screenshot of FacebookOAuth2UserInfo.java

HttpCookieOAuth2AuthorizationRequestRepository.java



Figure :Screenshot of HttpCookieOAuth2AuthorizationRequestRepository.java

1. Develop an application that utilizes an API.

Developed Application using React

* Auth-context.js

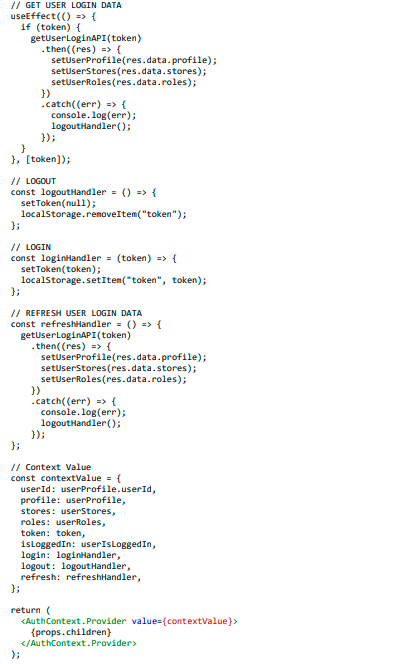


Figure :Screenshot of Auth-context.js

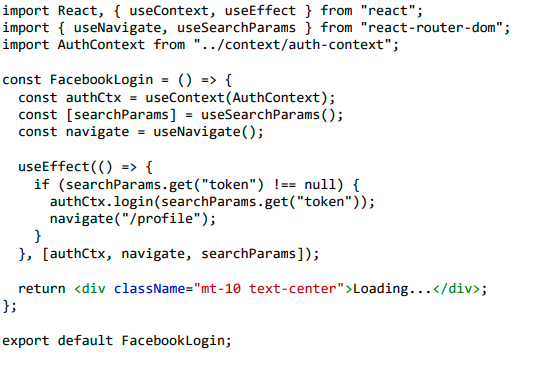
* FacebookLogin.jsx

Figure :Screenshot of FacebookLogin.jsx

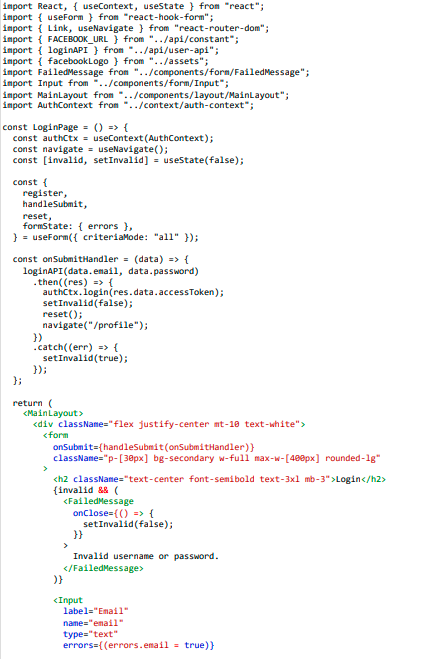
LoginPage.jsx

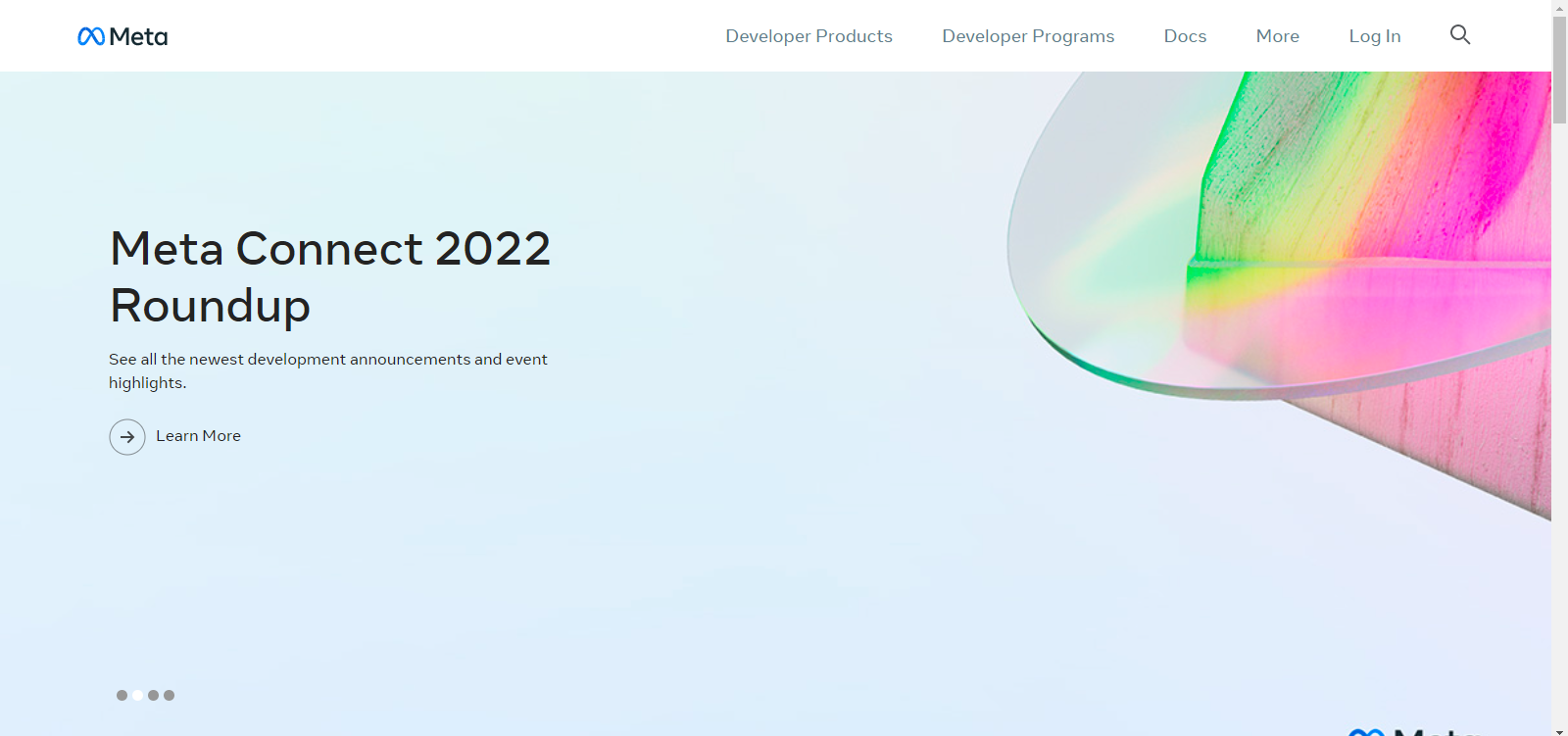


Figure :Screenshot of LoginPage.jsx

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

Using Facebook Login API

Step 1: Go to <https://developers.facebook.com/> and login



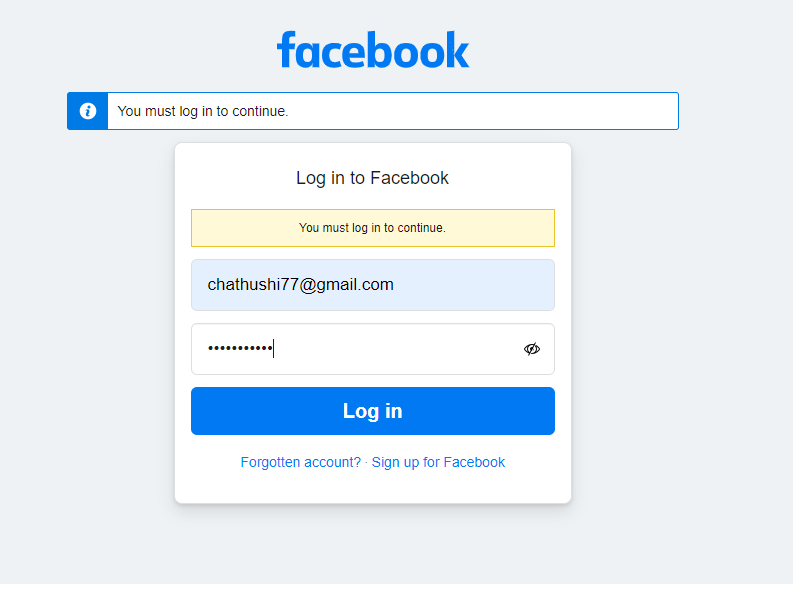


Figure : Application Construction using Facebook Login API step 1

**Step 2:** After login, you can see the following interface. You can click create app button and create app



Figure :Application Construction using Facebook Login API step 2

**Step 3:** Select type as “Gaming” and click next

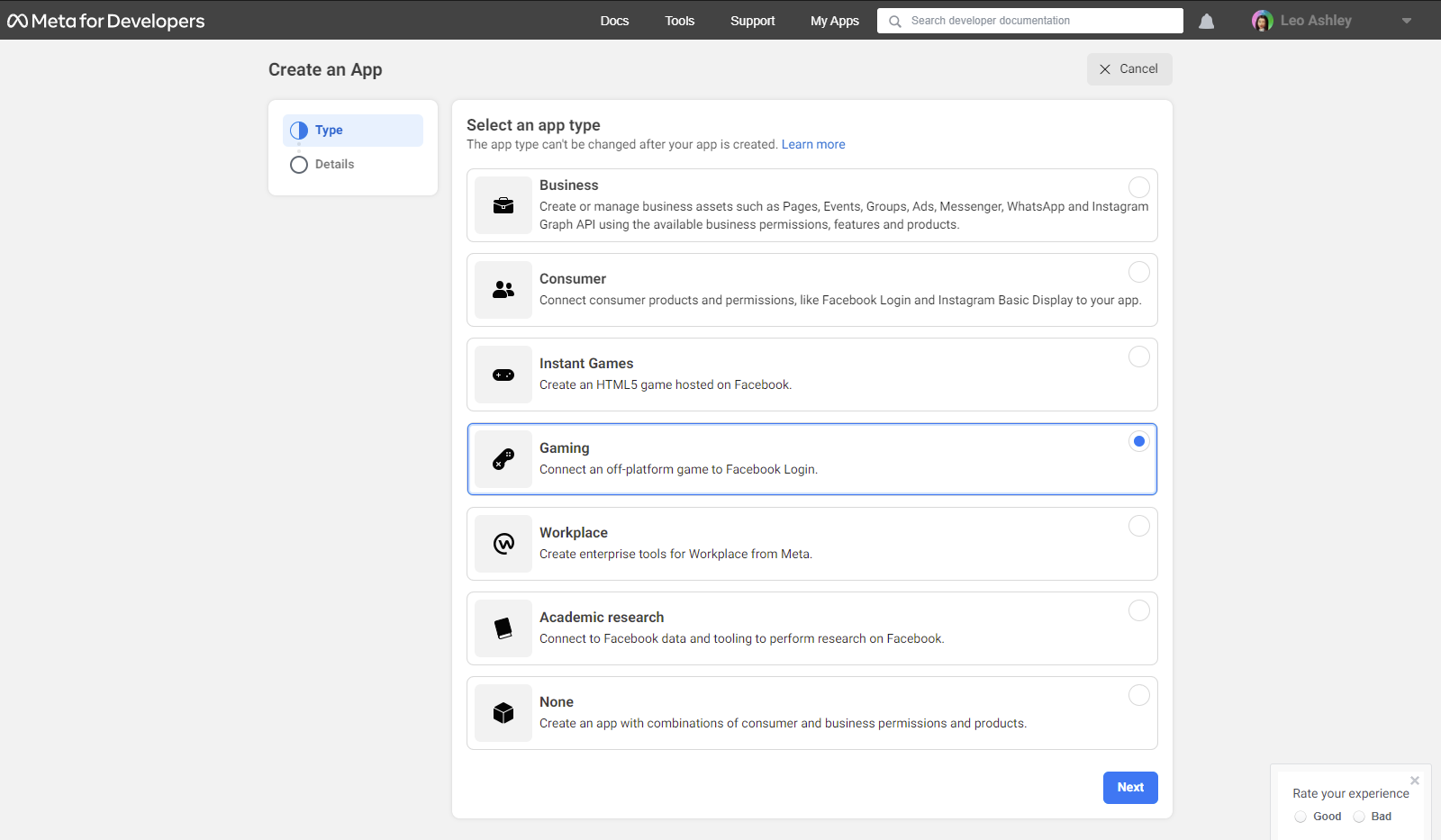


Figure :Application Construction using Facebook Login API step 3

**Step 4**: Give a name for app and contact email and click create app

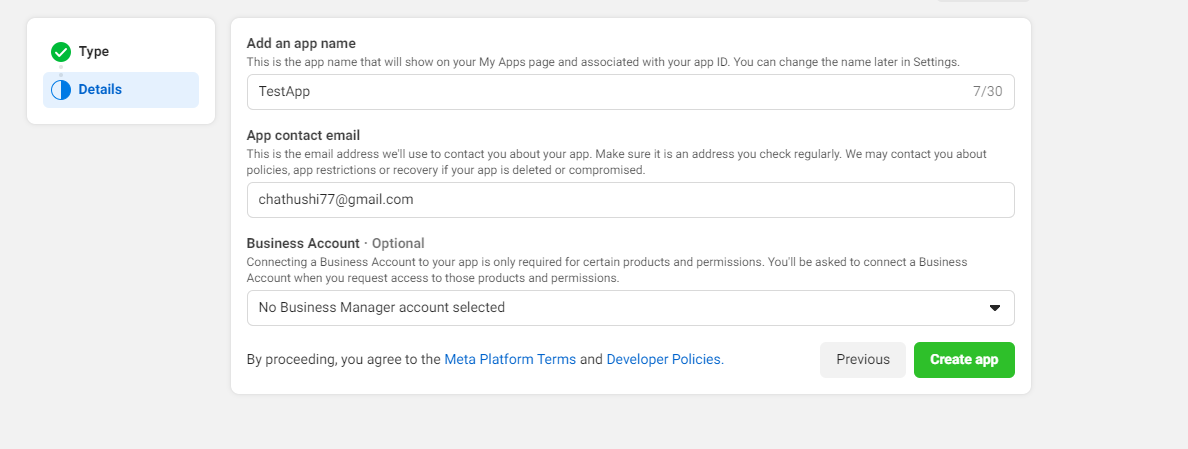


Figure :Application Construction using Facebook Login API step 4

**Step 5:** Copy the app ID and app secret for us and paste it on *application.yml*



Figure :Application Construction using Facebook Login API step 5

**Application.yml**

Figure : Application Construction using Facebook Login API (Application.yml)

Task 4

1. Implement white Box testing for the developed API of your Application
   1. What is white box testing?

White box testing is a testing approach wherein software program’s inner shape, design, and coding are tested to confirm enter-output flow and improve design, usability, and safety. In white box testing, code is visible to testers, so it's also known as clean container checking out, Open field testing, transparent container testing, Code-primarily based testing, and Glass box testing.

it's miles one in all two parts of the box trying out technique to software testing. Its counterpart, Blackbox testing, includes testing from an outside or quit-consumer attitude. alternatively, White box testing in software program engineering is primarily based on the internal workings of an software and revolves round inner testing.

* 1. Why we use white box testing?

White box testing ensures,

* Business requirement are satisfied
* Each entity of tech is perfectly integrated together
* Enable to search invisible bugs
* Ensure greater user experience

Testing APIs with Postman

Register

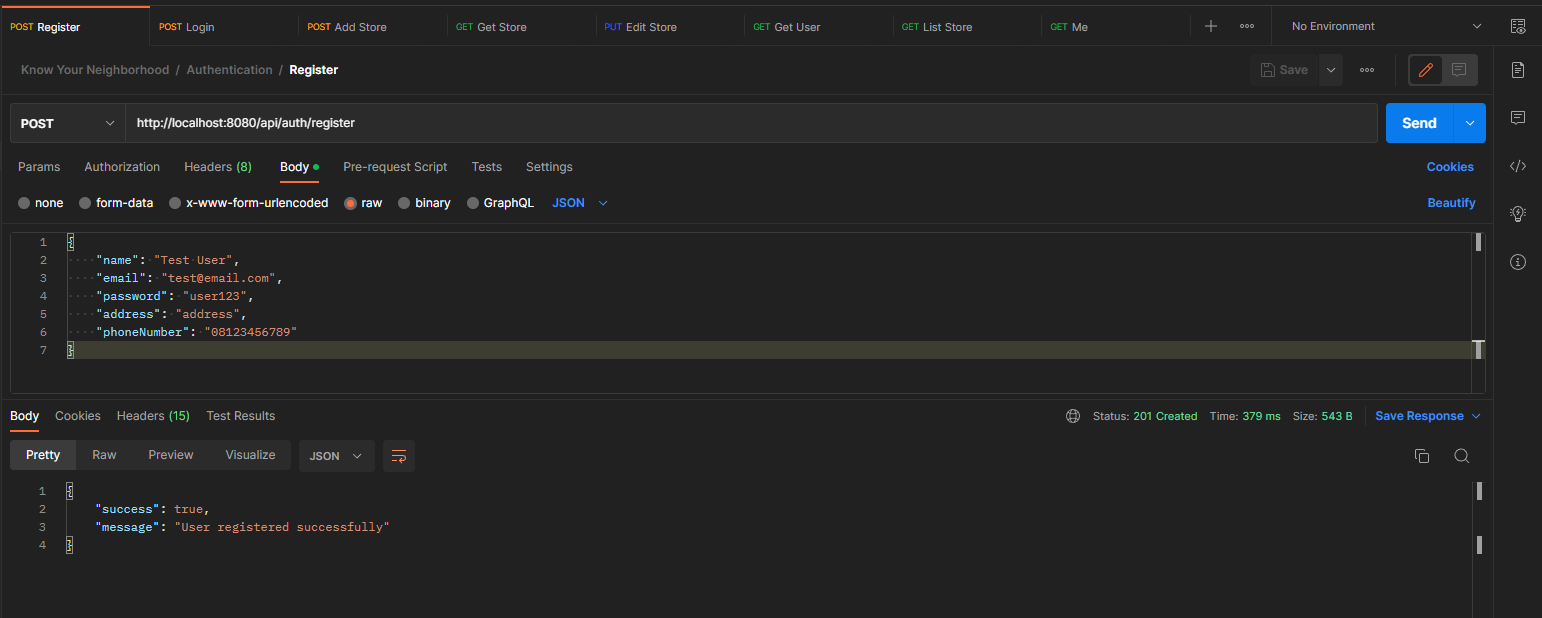


Figure : Postman Testing - Registration

Login

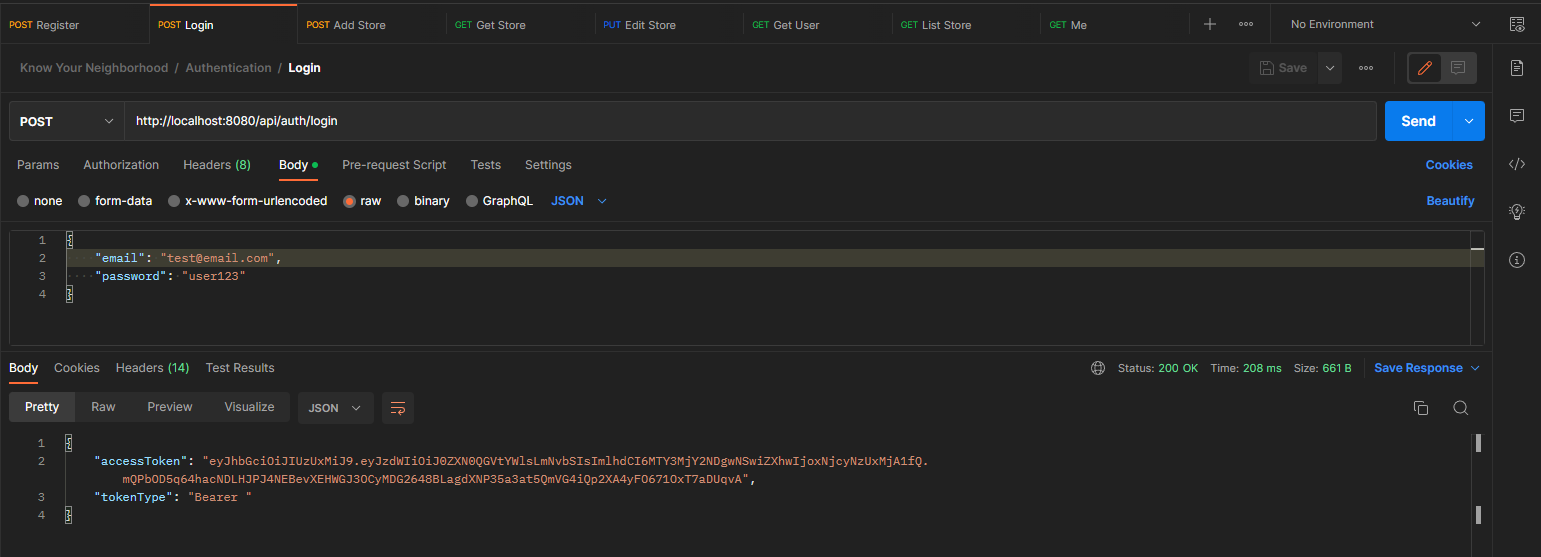


Figure :Postman Testing - Login

Get User Login

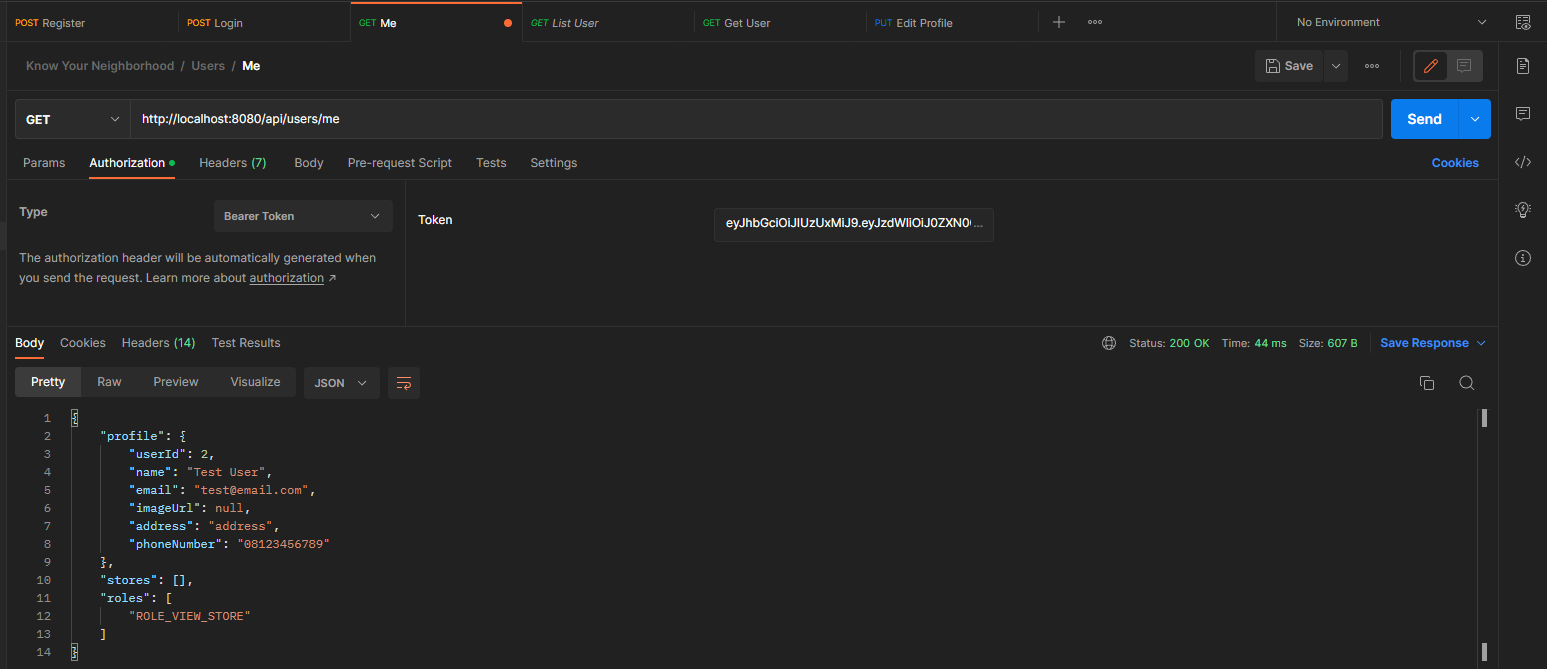


Figure :Postman Testing - Get user Login

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

Black box testing involves testing a machine with no prior knowledge of its internal workings. A tester gives input, and observes the output generated through the gadget underneath take a look at. This makes it possible to identify how the system responds to predicted and unexpected person movements, its response time, usability troubles and reliability issues.

Black container testing is an effective testing method because it physical activities a gadget give up-to-stop. much like cease-users “don’t care” how a device is coded or architected, and expect to acquire the perfect response to their requests, a tester can simulate consumer activity and notice if the system gives you on its promises. alongside the manner, a black field take a look at evaluates all relevant subsystems, which includes UI/UX, net server or application server, database, dependencies, and incorporated systems.

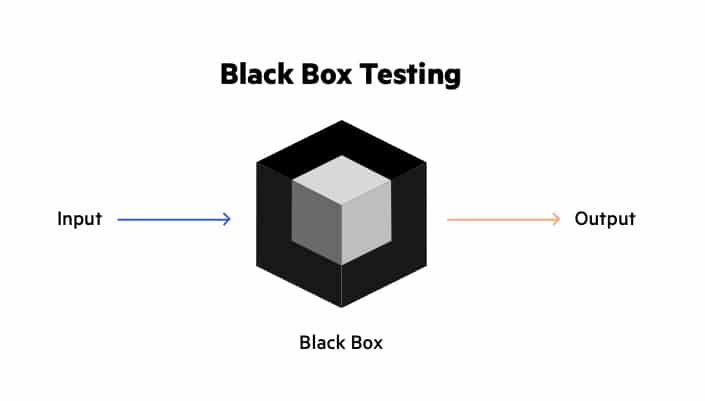


Figure :Black box testing

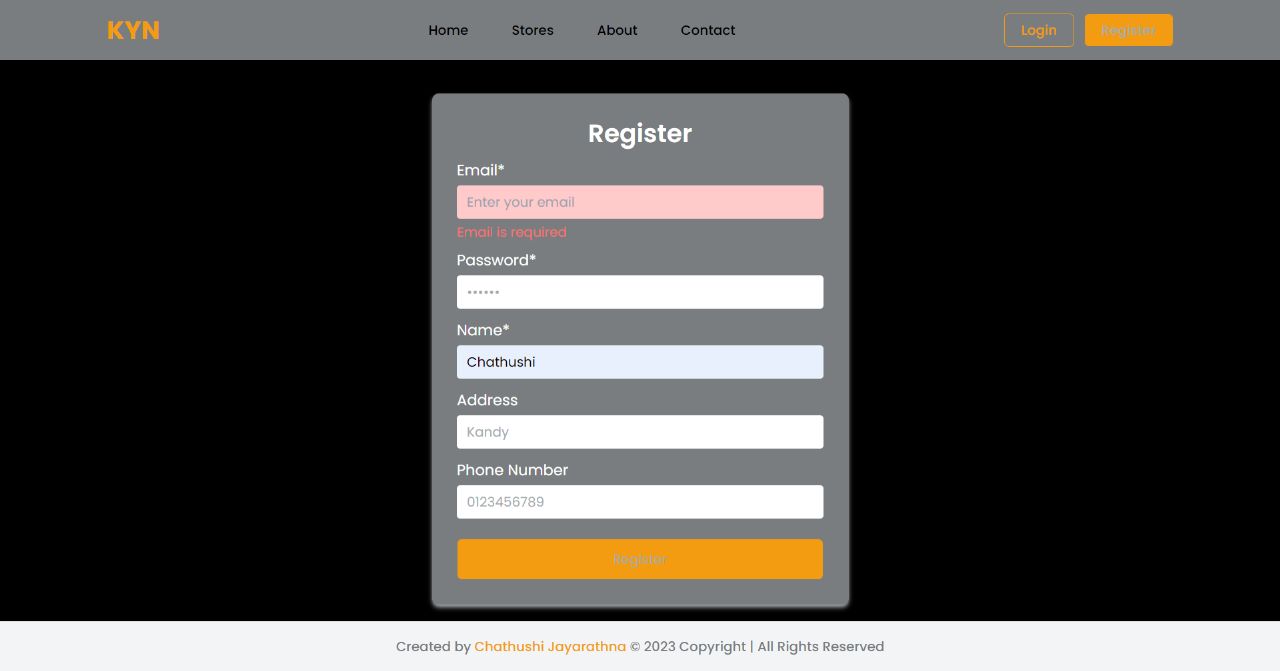
* 1. Benefits of black box testing
* The tester doesn’t need any technical expertise to check the machine. It is essential to recognize the user’s perspective.
* Testing is achieved after development, and both the sports are independent of every other.
* It works for a greater giant insurance which is typically neglected out by way of testers as they fail to see the larger photograph of the software.
* Test cases can be generated before improvement and right after specification.
* Black box test method is near agile.

Registration

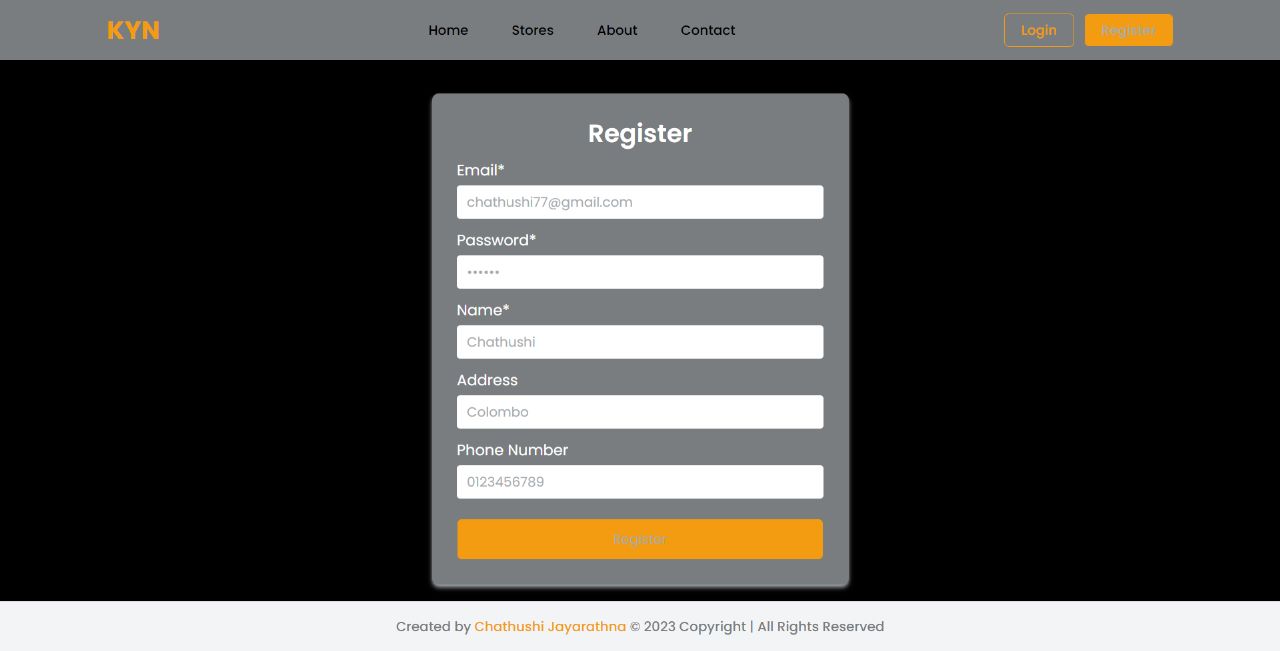
|  |  |
| --- | --- |
| **Test Scenario** | Registration |
| TS001 |
| **Test Cases** | Registration in the Know Your Neighborhood Website |
| TC001 | Validating data that is inpputed into registration form |
| TC002 | After user submitting correct data it should show a success message |

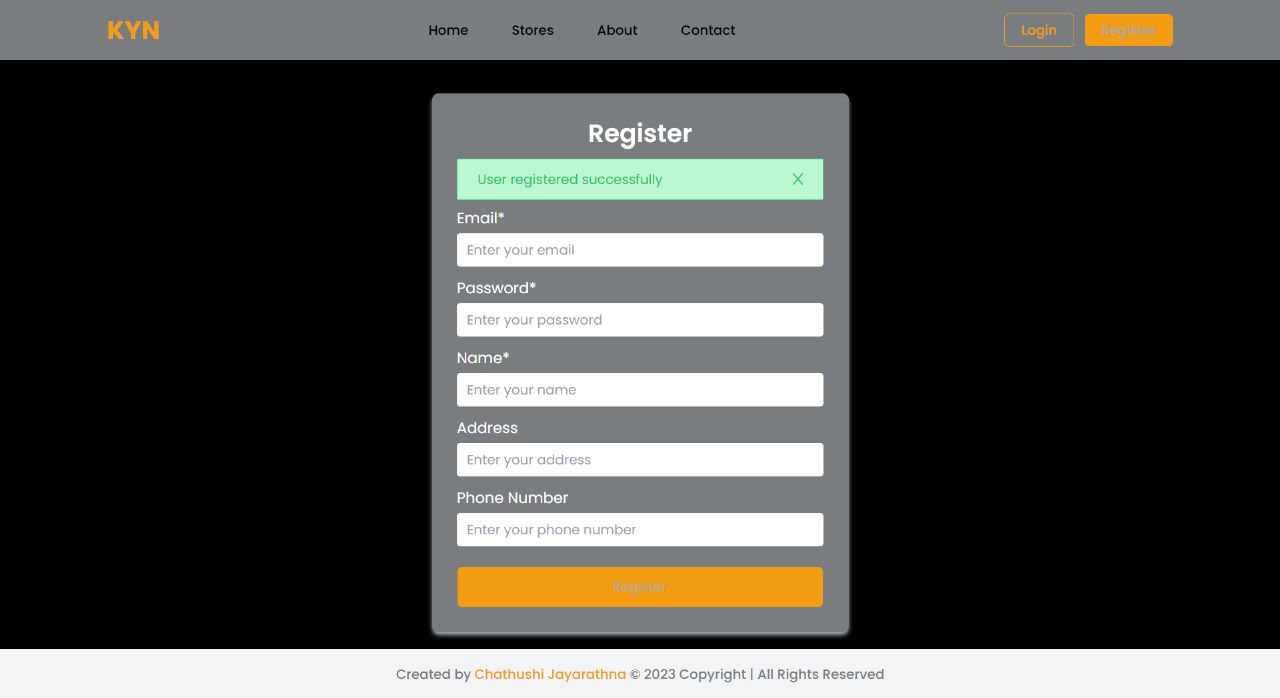
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test case | Scenario | Preconditions | Execution steps | Test Data | Expected Result | Actual Result | Pass/Fail | Reference Evidence |
| TC01 | Ensure that users can register on the KYN website | the website has been designed and developed | * Go to KYN Website * Go to login oage * Click on the register button * Fill in all input data * Click on Register button | email = “\_\_\_\_\_ ”  Password = 123456  Name = Chathu  Address = Kandy  Phone number = 0123456789 | If user enter blank data It should throw an error . | As expected, | PASS | TS001 – TC01 |
| TC0 | Ensure that users can register on the KYN website | the website has been designed and developed | * Go to KYN Website * Go to login oage * Click on the register button * Fill in all input data * Click on Register button | email = chathushi77@gmail.com  Password = 1234567  Name = Chathu  Address = Kandy  Phone number = 0123456789 | Show success message after submitting data . | As expected, | PASS | TS001 – TC02 |

**Fig: TS001 – TC01**



**Fig: TS001 – TC02**



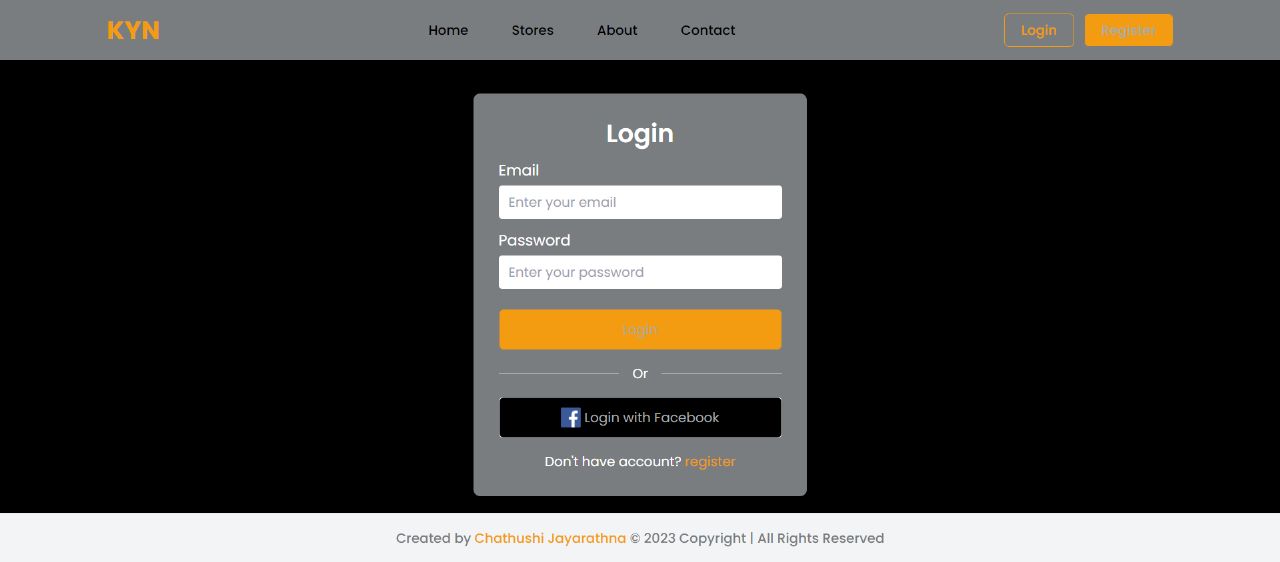


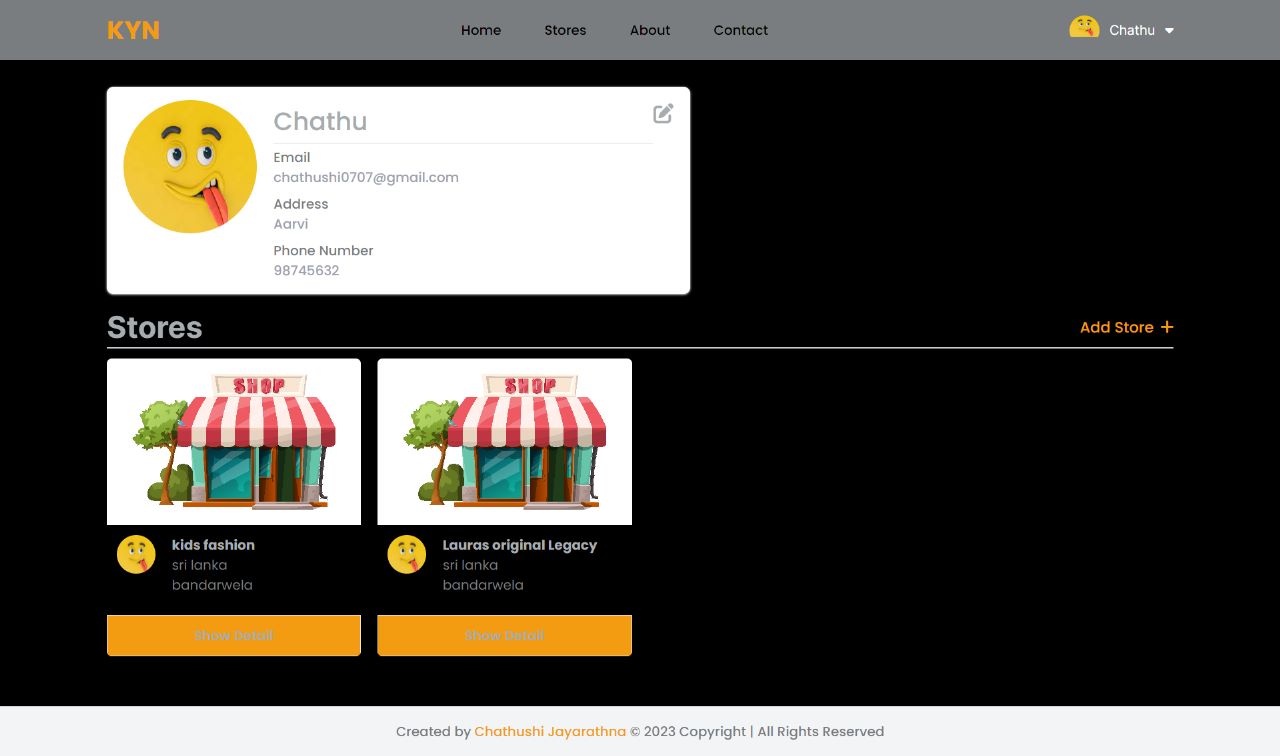
Login

|  |  |
| --- | --- |
| **Test Scenario** | Login |
| TS002 |
| **Test Cases** | Login in Know Your Neighborhood Website |
| TC001 | Login with local account |
| TC002 | Login with facebook API |

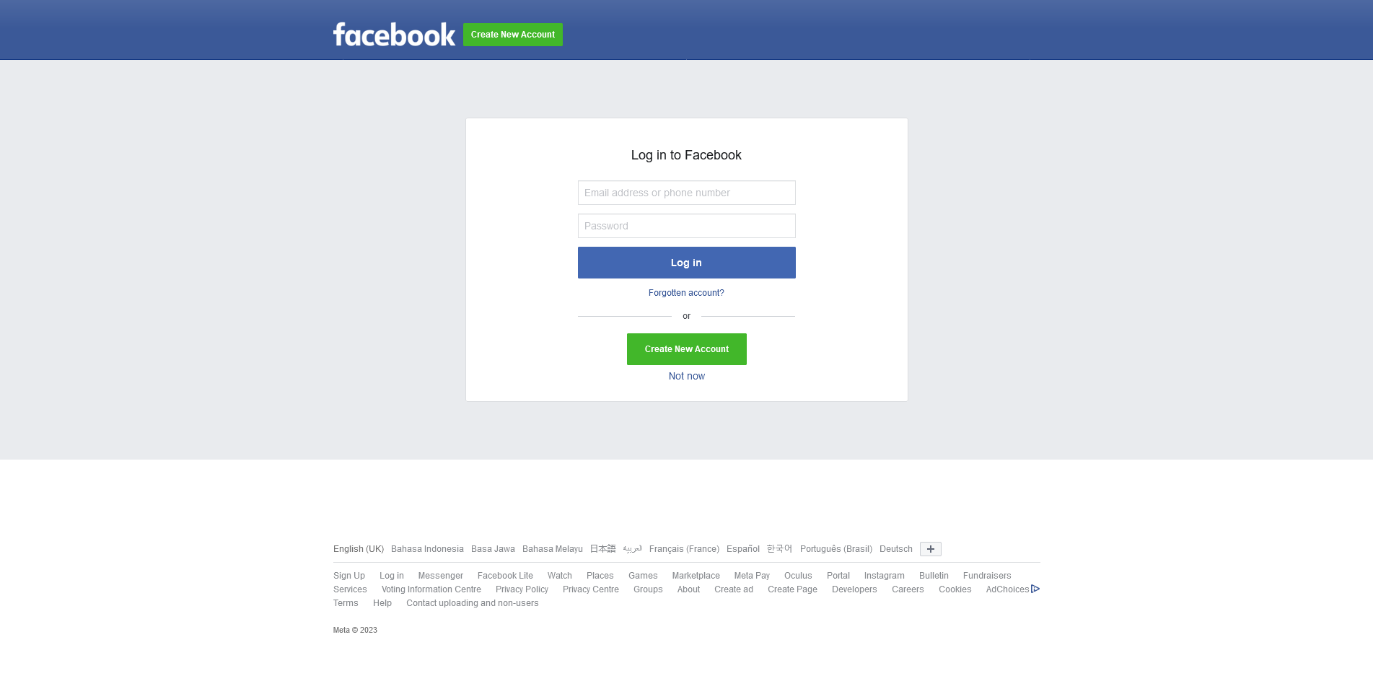
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test case | Scenario | Preconditions | Execution steps | Test Data | Expected Result | Actual Result | Pass/Fail | Reference Evidence |
| TC01 | Ensure that users can register on the KYN website | the website has been designed and developed | * Go to KYN Website * Go to login oage * Click on the register button * Fill in all input data * Click on login button | Email = [chathushi77@email.com](mailto:chathushi77@email.com)  Password = 123456 | After user entering valid data, user should be redirected to their profile page . | As expected, | PASS | TS002 – TC01 |
| TC0 | Ensure that users can register on the KYN website | the website has been designed and developed | * Go to KYN Website * Go to login oage * Click on the register button * Fill in all input data * Click on login button | “Valid facebook account”0123456789 | After user entering valid data, user should be redirected to their profile page | As expected, | PASS | TS002-TC01 |

**Fig: TS002 – TC01**





**Fig:TS002- TC02**

****

**Task 5**

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

**REST API**

**Strengths of REST API**

* **Flexible across languages and frameworks**

One obvious reason why the rest API has been so successful is that it really works across languages and frameworks. No matter how developers build their programs, a rest API will work for them.

* **High interpretability**

The fact that rest APIs are URL-primarily based, in which every URL gives get right of entry to a certain amount of data, renders rest APIs extraordinarily easy to interpret. All a third-birthday party developer calls for is a URL, knowledge of what statistics they will discover there, and knowledge of the way they are able to engage with that information.

* **Server-side logic**

For an agency constructing a rest API, its URL-pushed nature is extraordinarily simple to work with. The server-aspect builders actually define a URL in which they want positive information to be seen, write common sense for the way to cope with every form of request, and the endpoint is created. However, as an employer's API scales in utilization, identifying the ideal manner to structure endpoints and the statistics they offer increases in complexity, a relative weakness of this architecture, transitioning to our next phase.

**Weakness of REST API**

* They may be less at ease than different types of APIs, as they rely upon the underlying delivery protocol (HTTP) to offer protection.
* They may be vulnerable to mistakes if no longer implemented well, as they rely on clients to offer accurate entries.
* They can be less efficient than different kinds of APIs, as they require extra overhead (e.g. parsing JSON or XML) to manner facts.
* Few use instances may additionally require extra complex features than the rest can provide.

1. Provide data security report of your developed application.

Spring Security

Spring security is a powerful and incredibly customizable authentication and access-manage framework. it is the de-facto standard for securing Spring-primarily based applications.

Spring security is a framework that specializes in offering each authentication and authorization to Java packages. like several Spring tasks, the real power of Spring security is discovered in how effortlessly it is able to be prolonged to fulfill custom requirements

Features

* Comprehensive and extensible support for both Authentication and Authorization
* Protection against attacks like session fixation, clickjacking, cross site request forgery, etc.
* Servlet API integration

Authentication with JSON Web Token (JWT)

Authentication with JSON net Token (JWT) is a famous technique for securing web packages. There are several motives why this project chooses to apply JWT for authentication:

* JWTs are self-contained: A JWT contains all the records necessary to authenticate a single user, compact token. This eliminates the want to shop authentication statistics in a consultation or at the server, which can simplify the implementation of the application.
* JWTs are at ease: JWTs are signed with a secret key, which makes it difficult for attackers to forge them or tamper with their contents.
* JWTs are broadly supported: JWT is a trend that is supported with the aid of a huge variety of platforms and libraries, making it clean to combine 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.

**Task 6**

1. **Developed Pages**

Home Page

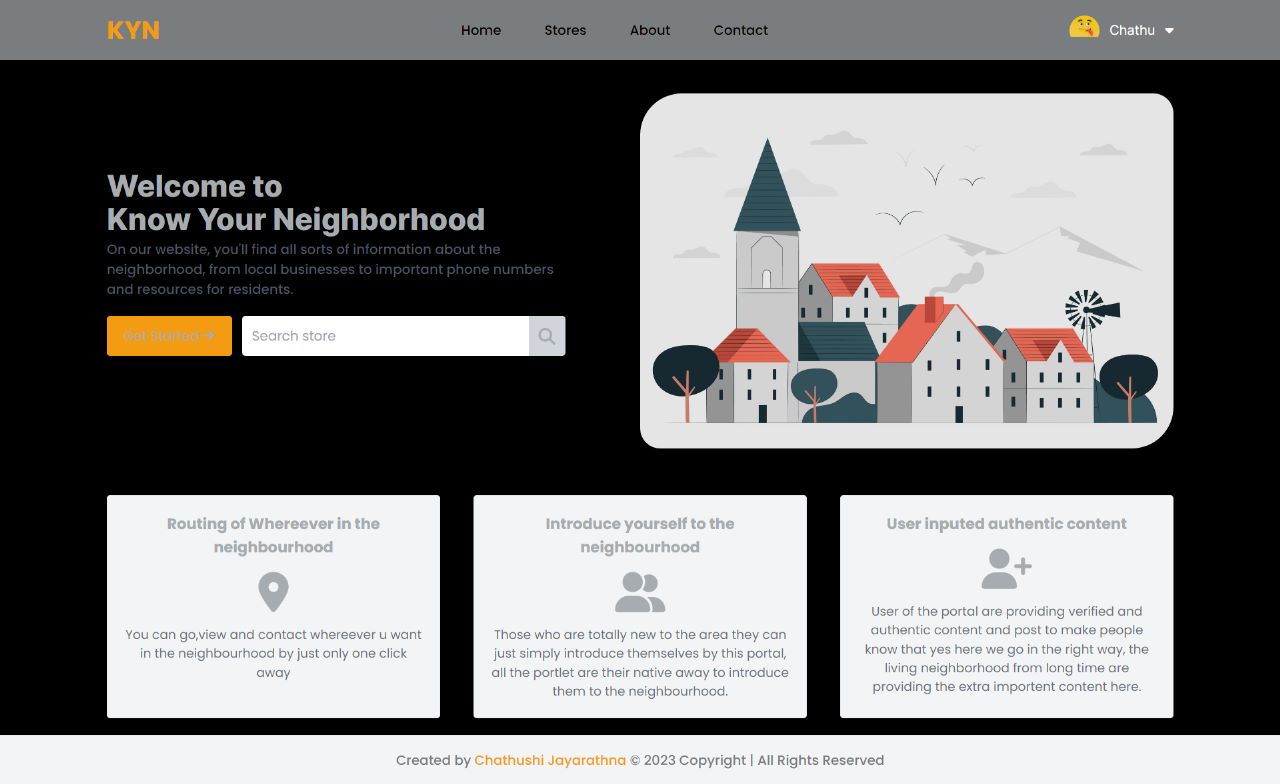


Figure :Screenshot of Home Page

Login Page

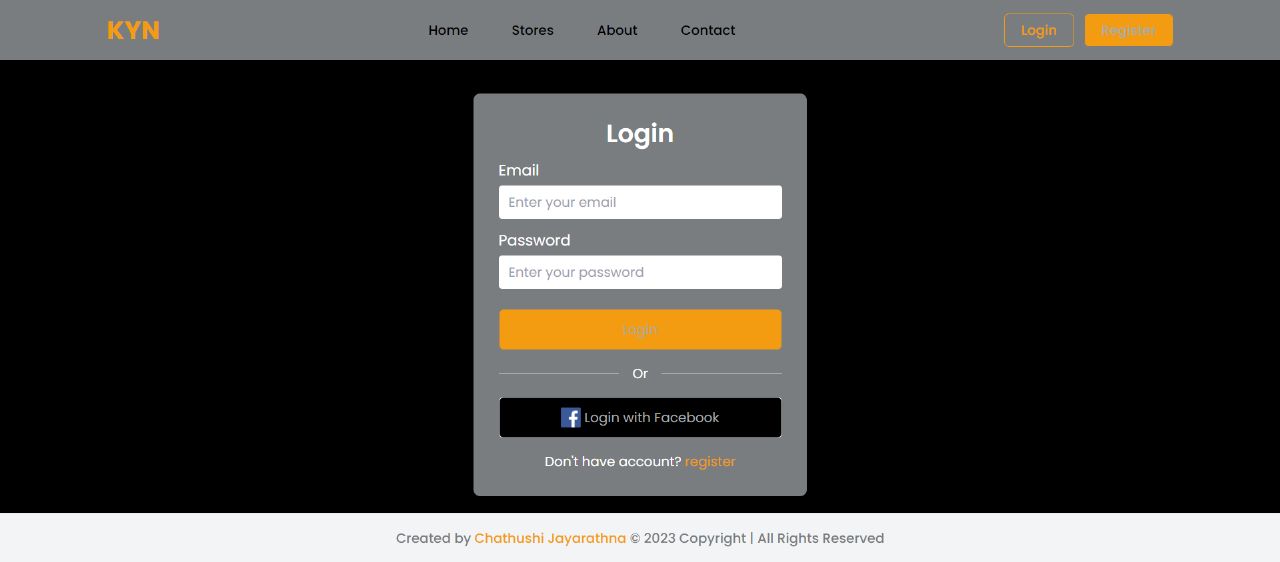


Figure :Screenshot of Login Page

Facebook Login

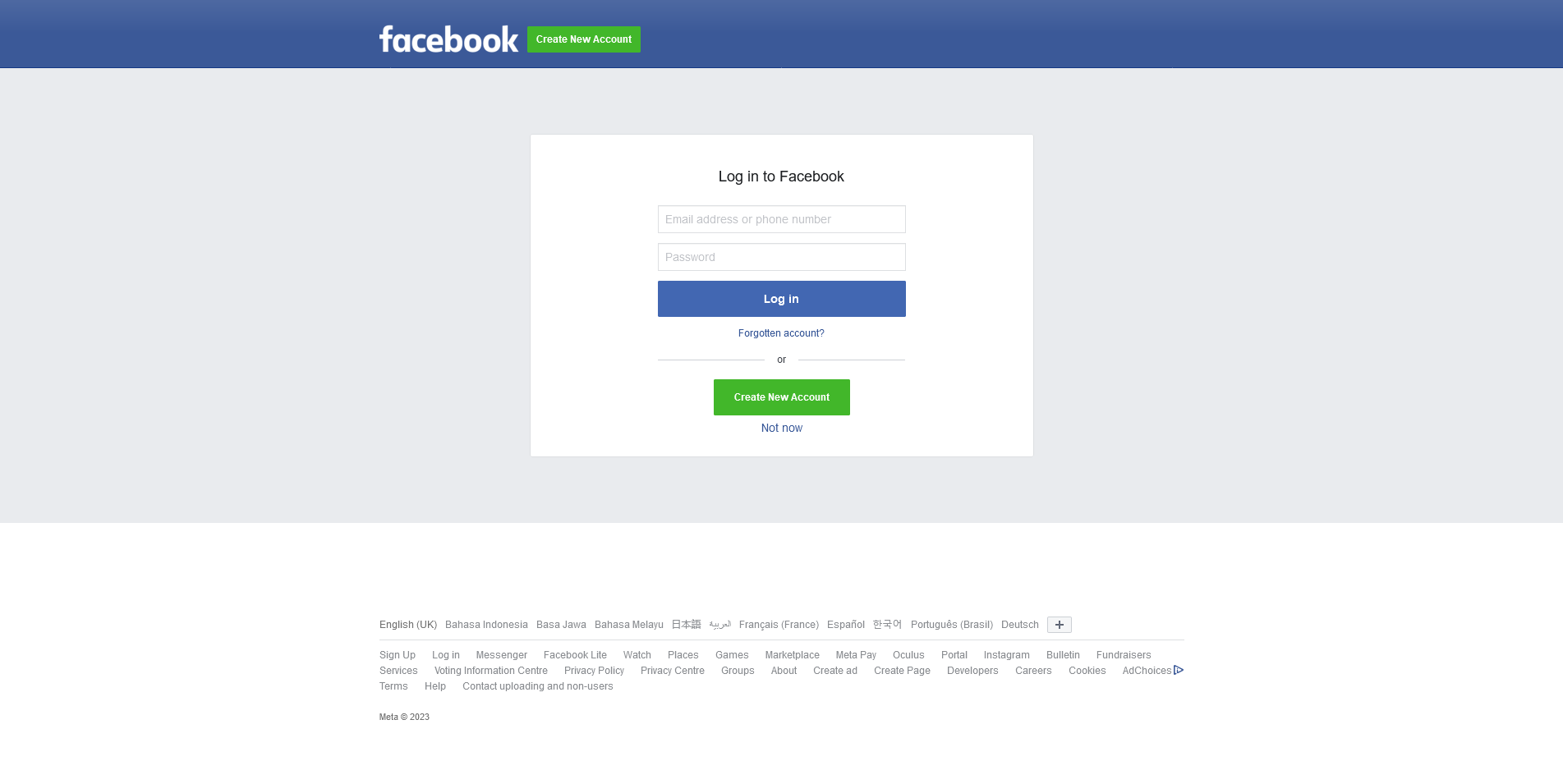


Figure :Screenshot of Facebook login

Register

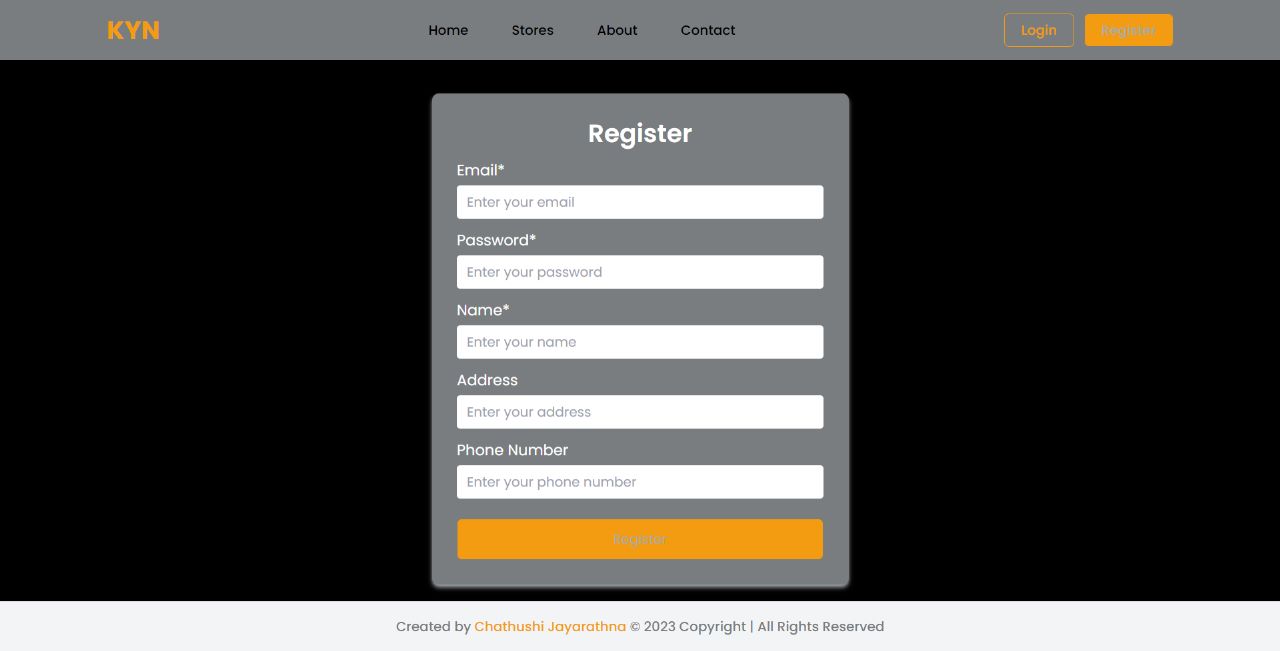


Figure :Screenshot of Registration

Stores

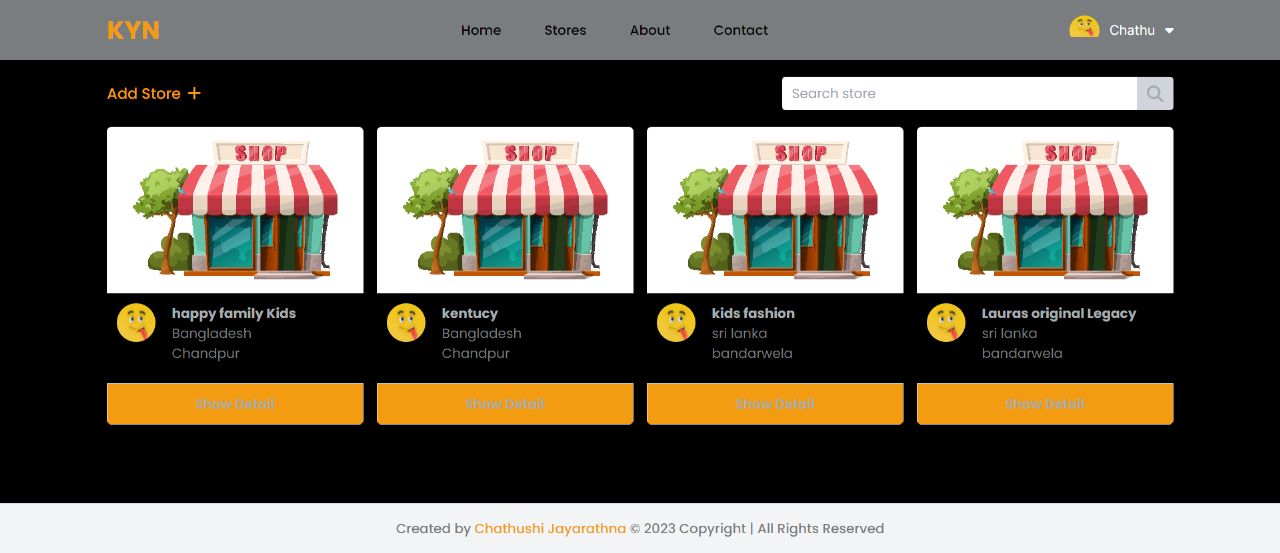


Figure :Screenshot of Stores

Store Detail

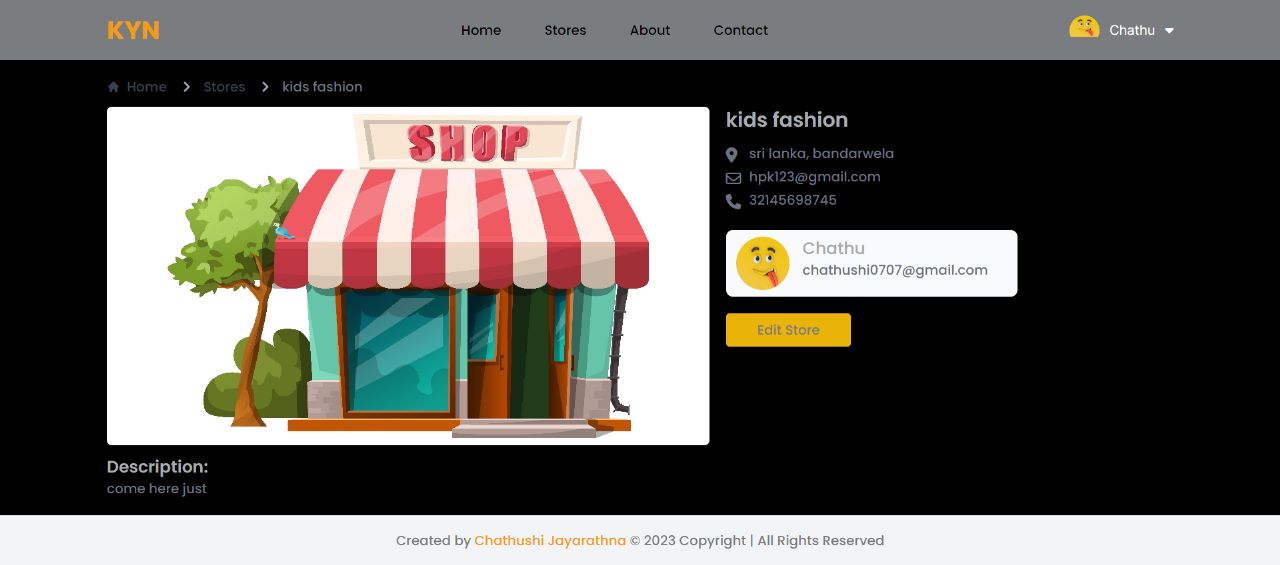


Figure :Screenshot of Screenshot Store Details

Add Store

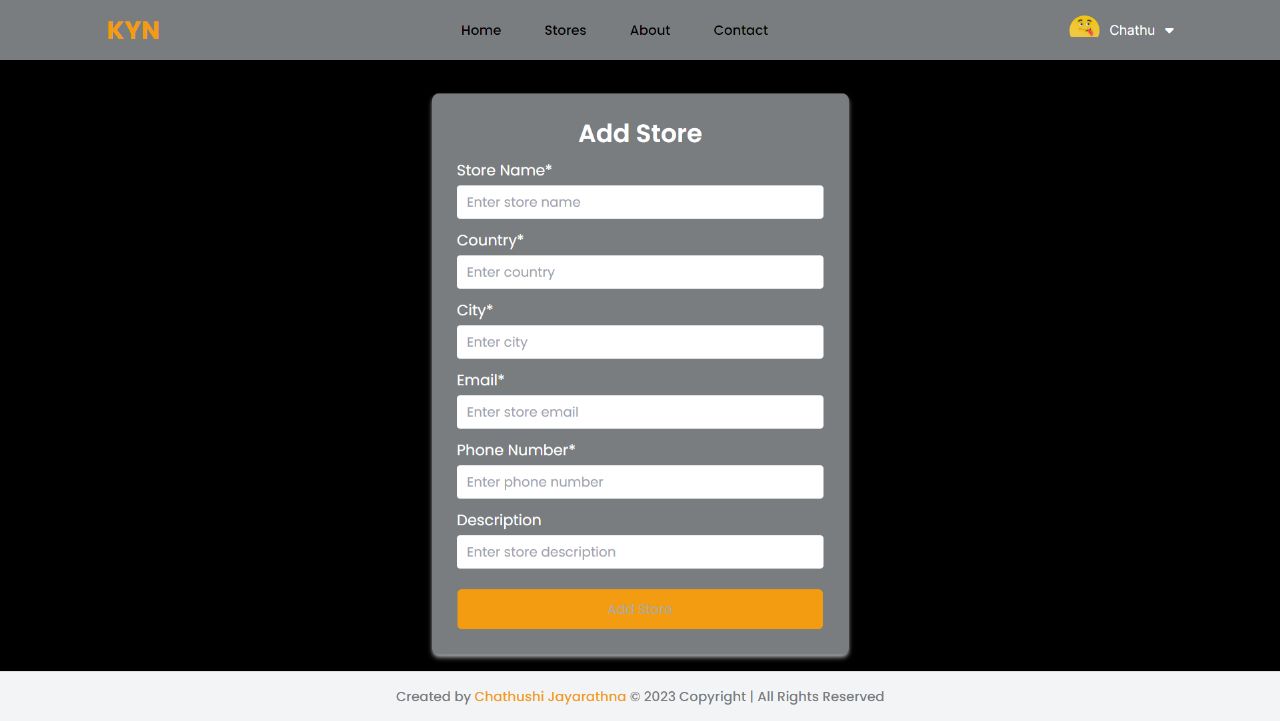


Figure :Screenshot of Add Stores

**Edit Store**

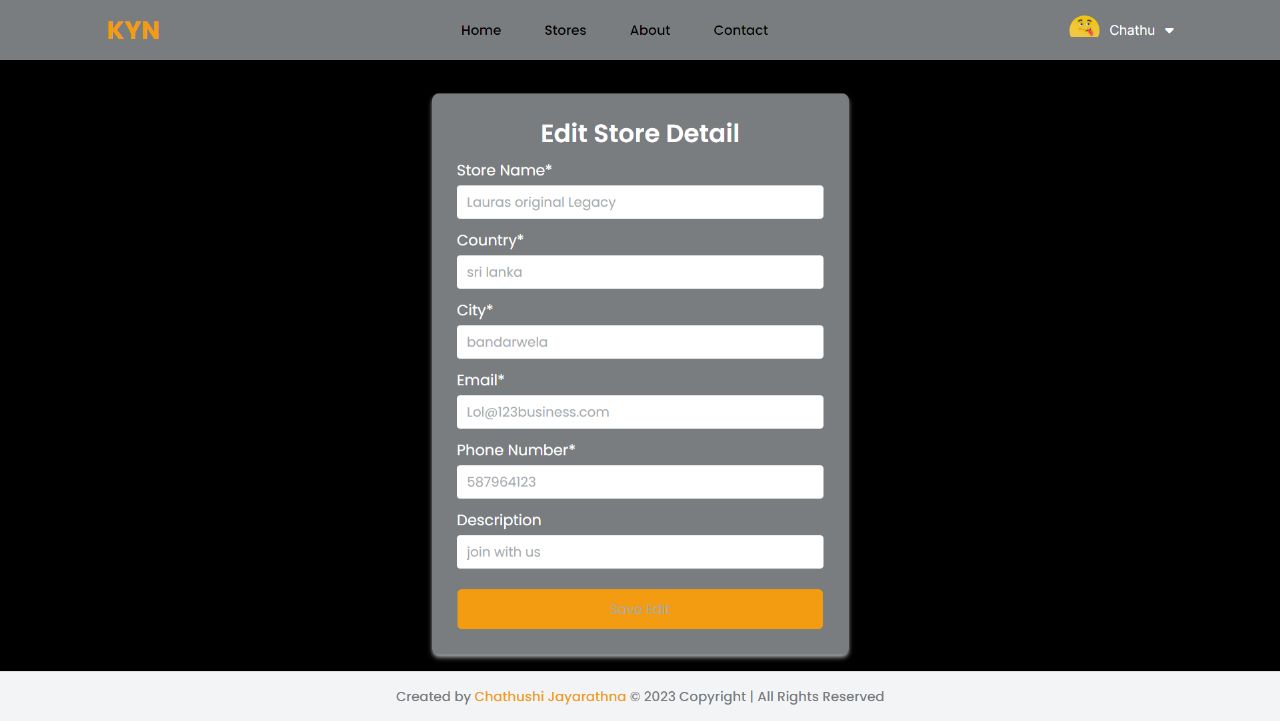


Figure :Screenshot of Edit Store

Profile Page

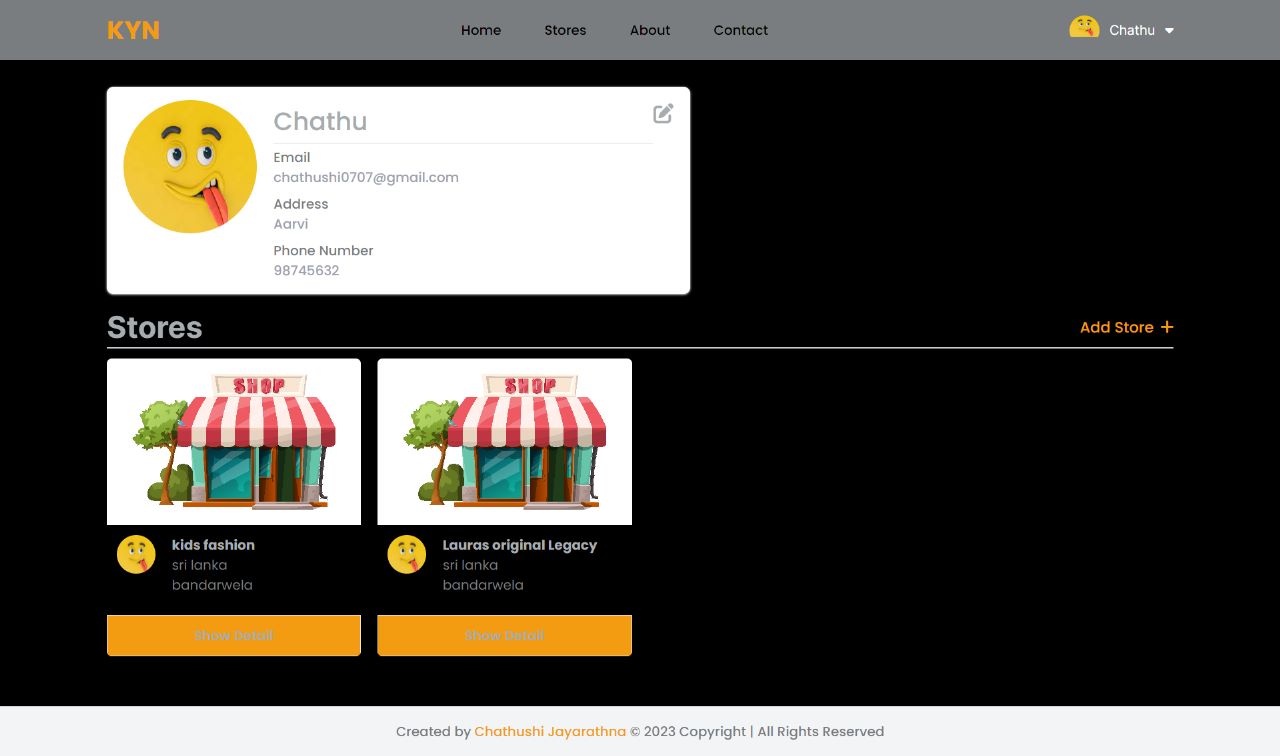


Figure :Screenshot of Profile Page

Edit Profile

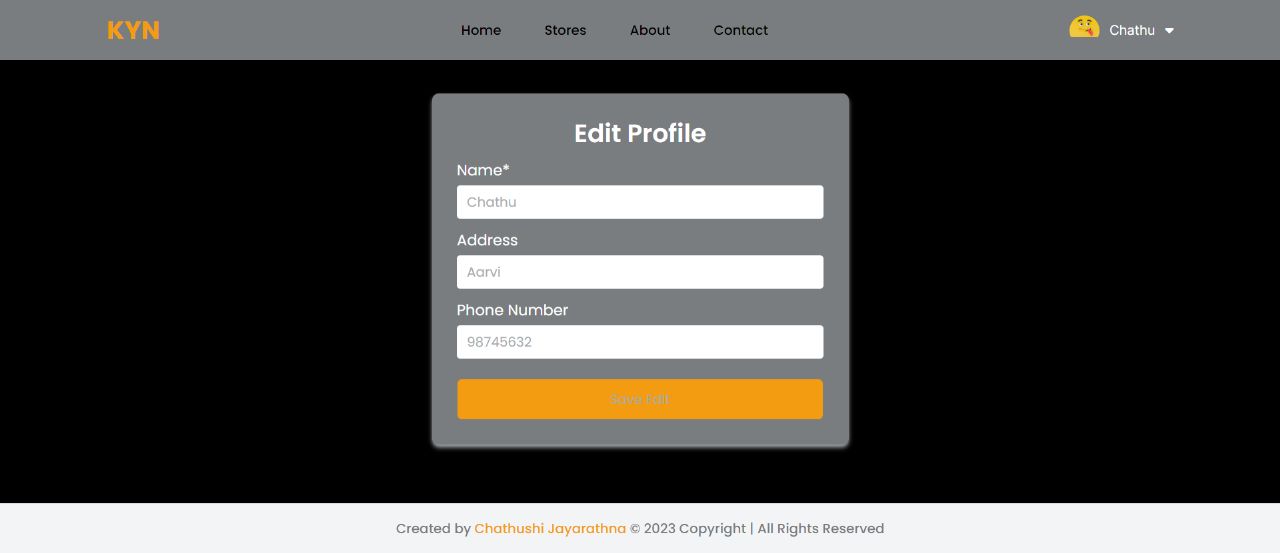


Figure :Screenshot of Edit Profile Page

**About Us**

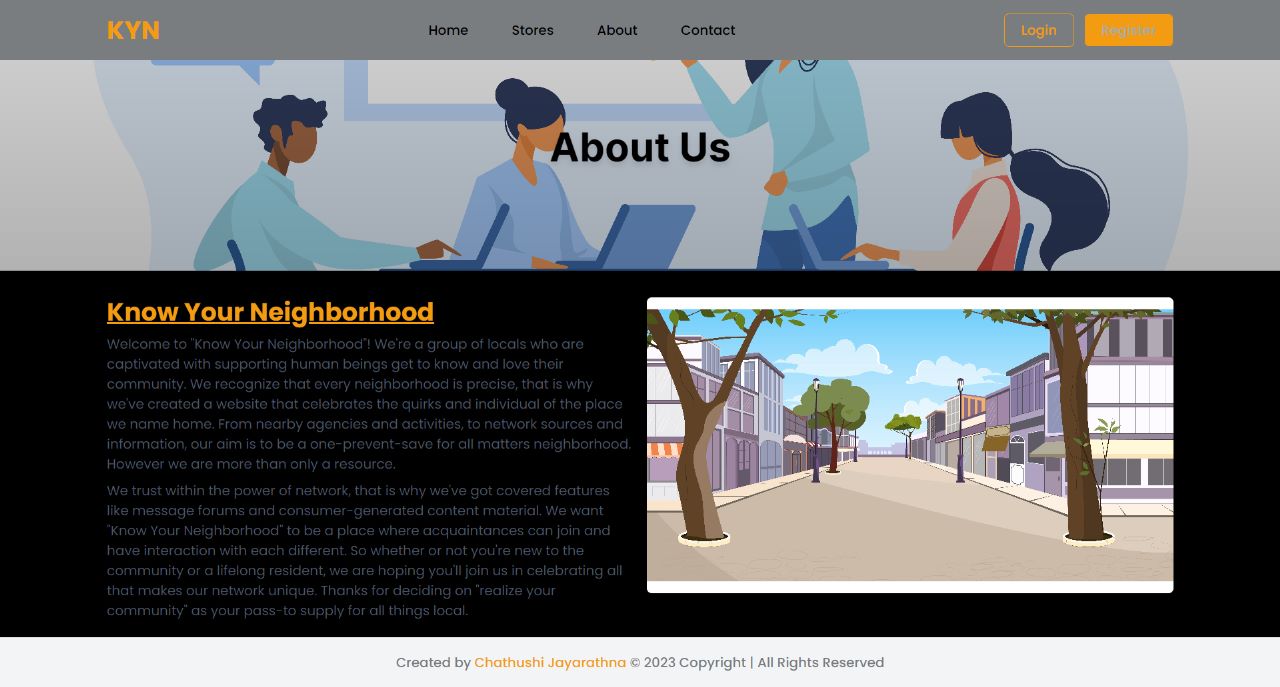


Figure :Screenshot of About Us Page

**Contact Us**

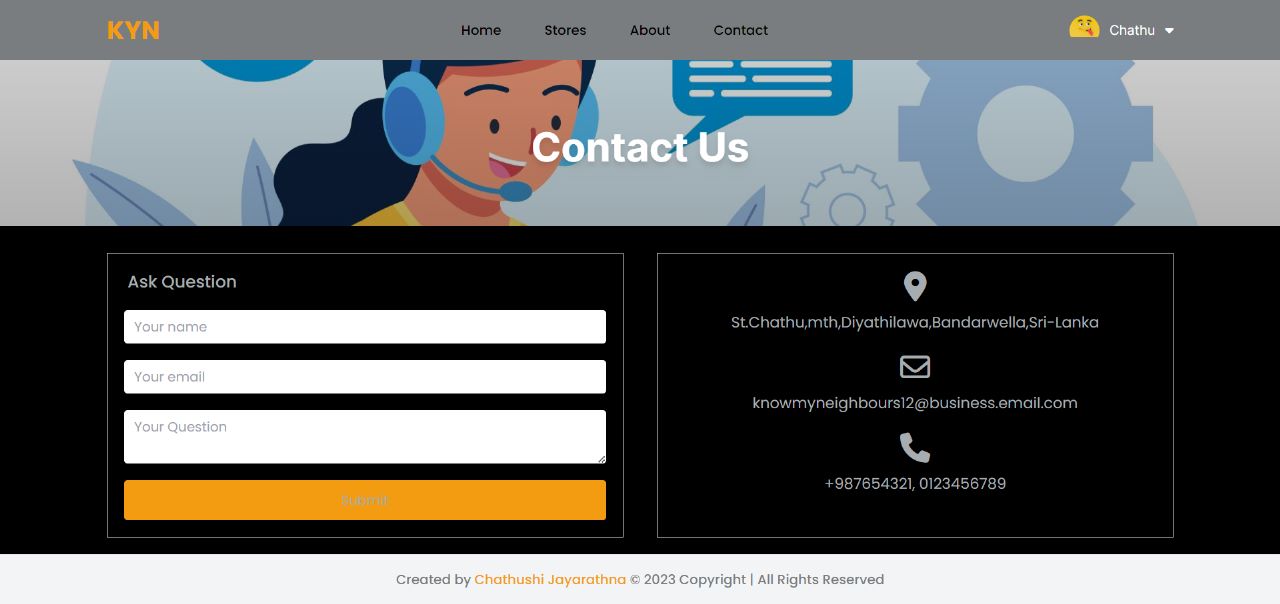


Figure :Screenshot of Contact Us Page