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

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name** | | **Assessor name** | |
| Agung Yuda Pratama | |  | |
| **Date issued** | **Completion date** | | **Submitted on** |
|  | June 2023 | | June 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: 06 June 2023 |

1. Project Background

The Know-Your-Neighborhood application was developed using Spring Boot.

This assignment gives you an opportunity to demonstrate your capabilities in the following

areas:

• 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

1. Project Objective

• The project’s objective is to analyze various current APIs for each of your examples, assess each one's acceptability, look for any potential security holes, and establish a login with the chosen API on an already-existing website.

* Tools and Platform

1. Visual Studio Code
2. React JS
3. Facebook API
4. Spring Suite Tools
5. Postman
6. MySQL Workbench
7. Axure RP
8. Google Chrome
9. Microsoft Word
10. 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:

1. Home Page
2. Registration Page
3. Login Pages with API link
4. Contact us Page
5. About us Page
6. Terms and Conditions Page
7. 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. An interface that can link one program to another is the API itself. In other words, the API serves as a bridge between various apps that run on the same or distinct platforms. Furthermore, several programming languages can be communicated with using the API itself. Certainly simple for devs to do. Actually, developers may utilize the API to obtain the data they require from other platforms, negating the need to supply all the information themselves.

An illustration of API that can be used is that of a waiter in a restaurant. The waiter's role in a restaurant is to introduce customers to the chef. As a result, customers just place food orders from the menu list, and the waiter notifies the cook. After some time, the waiter will return to the guest with the prepared dishes.

1. Its Role

The purpose of an API is to give applications a standardized method of requesting resources or services from other software components without the requirement for them to comprehend the intricate details of how those components operate inside. It serves as a middleman that enables information exchange and action execution between various software systems and facilitates integration.

1. Need for API

API allows different software components to communicate and transfer data. This allows developers to bridge the gap between small, discrete pieces of code to build robust, robust, and secure applications that meet user needs. APIs enhance collaboration by promoting interoperability, enabling modular development, encouraging specialization, facilitating third-party integration, accelerating development, and supporting service composition. By leveraging APIs, teams can work together effectively, seamlessly integrate systems, and create robust software solutions by combining the strengths of multiple parties.

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

Here are some examples of APIs for mobile, desktop and web APIs:

* Mobile APIs:
  + Google Maps API
  + Spotify Web API
  + GIPHY API
* Desktop APIs:
  + PayPal API
  + Windows API
* Web APIs:
  + Google Maps API
  + Google Calendar API
  + YouTube Data API
  + Facebook API
  + Twitter API

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

SDK (Software Development Kit) is a collection of software development tools a platform or company provides. An SDK often consists of many parts, including APIs, libraries, documentation, and other tools that help programmers create applications for a specific platform or environment. The SDK makes it simple for programmers to utilize the capabilities offered by the platform or business.

API defines the interface and protocols for interacting with a software system. At the same time, an SDK is a comprehensive package that includes the API and additional tools, libraries, and resources to simplify development for a specific platform or system. The API is a core component of an SDK, providing the means to access the functionality, while the SDK extends the API by offering a complete set of development tools and resources.

An SDK may have one or more platform-specific APIs depending on the situation. To interface their apps with pre-existing platforms or systems and shorten the application development cycle, developers employ SDKs and APIs. An SDK typically includes one or more APIs that developers use to access and utilize the functionality offered by a platform, developer and business, which is the relationship between an API and an SDK.

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

Public API is often referred to as Open API. As the name implies, Public API is an API that anyone can use across platforms.

Also, this type of API is the easiest to use. You can sign up or directly take advantage of it in your application. For example, the API for Google Maps.

* + 1. Private API

Unlike the public API, Private API is an API not open for public use. Usually, this type of API is created for internal purposes in developing certain applications.

For example, the API from the back end is used to access the front end of a website. Or applications for mobile application development.

* + 1. API Partners

API partners can be used for public purposes, but only those with permission can use them. As with public APIs, you must register with the API provider first. Then, use it only in certain applications according to the agreement. For example, the Pinterest API.

* + 1. Composite API

Composite API is an API that stores data from various servers or hosts in one place. Of course, this really saves time for the user. That's because users can get various data types in just one access.

1. Examine the uses of APIs for a particular type

APIs have many uses across different industries and applications. For example, mobile app developers use APIs to integrate their apps with social media platforms such as Facebook and Twitter. E-commerce companies use APIs to integrate their online stores with payment gateways such as PayPal and Stripe. Healthcare providers use APIs to integrate electronic health records (EHRs) with other healthcare systems.

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

* Injection Attack: APIs that do not properly validate and sanitize user input can be vulnerable to injection attacks, where malicious code or commands are injected into API requests or responses.
* DDoS Attacks: Distributed Denial of Service (DDoS) attacks, in which numerous requests overload an API's resources and disrupt service, can target poorly protected APIs.
* Broken Authentication: APIs that have weak or flawed authentication mechanisms can be exploited by attackers to gain unauthorized access to sensitive data or perform unauthorized actions.
* Sensitive Data Exposure: APIs that transmit or store sensitive data without proper encryption or protection mechanisms are at risk of exposing that data to unauthorized parties.
* Broken Access Control: APIs that do not enforce proper access controls and authorization checks can allow unauthorized users to access or manipulate sensitive data or perform actions they shouldn't have access to.
* Parameter Tampering: APIs that do not validate and enforce integrity checks on input parameters are susceptible to parameter tampering, where attackers modify parameters to manipulate or bypass business logic or gain unauthorized access.
* Man-in-the-Middle Attack (MITM): MITM attacks, in which an attacker intercepts and alters the data sent between the API and client, can be used against APIs that transport data over insecure networks or do not employ sufficient encryption and certificate validation.

1. Evaluate potential security issues in a suitable API of Know-Your-Neighborhood

In the scenario of the "Know-Your-Neighborhood" application, which provides information about neighborhoods, local amenities, and safety ratings, the following security issues may be relevant:

1. Injection: The API should ensure that any user-generated input is properly validated, sanitized, and parameterized to prevent SQL, command, or code injection attacks. This is crucial to protect against attempts to manipulate or compromise the underlying database or system.
2. Broken User Authentication: The API must have robust authentication mechanisms to prevent unauthorized access to user accounts or sensitive information. It should implement secure password storage, strong authentication protocols (e.g., multi-factor authentication), and protection against brute-force attacks.
3. Broken Access Control/Authorization: The API should enforce proper access controls to ensure that users can only access the data and perform actions that they are authorized for. It should validate user permissions and implement role-based access control (RBAC) or other suitable authorization mechanisms.

* Suitable API for the given scenario

The Public API, because we will use the RESTful API and Facebook API

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.**
2. Identify what are the requirements in “Know-Your-Neighborhood” project.

• 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

1. Select a suitable login API among 3 different login APIs in research in the project scenario.

* Facebook Login API

Users can log on using their Facebook credentials. Due to the prevalence of Facebook accounts among users, this makes for an efficient login process. Then, since Facebook has gained popularity before other programs, it is simpler for us to create an account using the Facebook API. We may thus assume that there are many Facebook users and that using this API will be quite profitable.

* Google OAuth2

Google OAuth2 allows users to authenticate using their Google account. It offers a secure login option. Because even now, Google has grown very fast and there are so many users worldwide, we believe using Google Login will help users ease their login process into the system.

* LinkedIn

Users can authenticate with their LinkedIn account using LinkedIn's login API. This API is appropriate if the project aims to reach professionals or business-minded people likely to have a LinkedIn account. Access to professional profile information is made possible through this, which may be helpful for the particular project functionality.

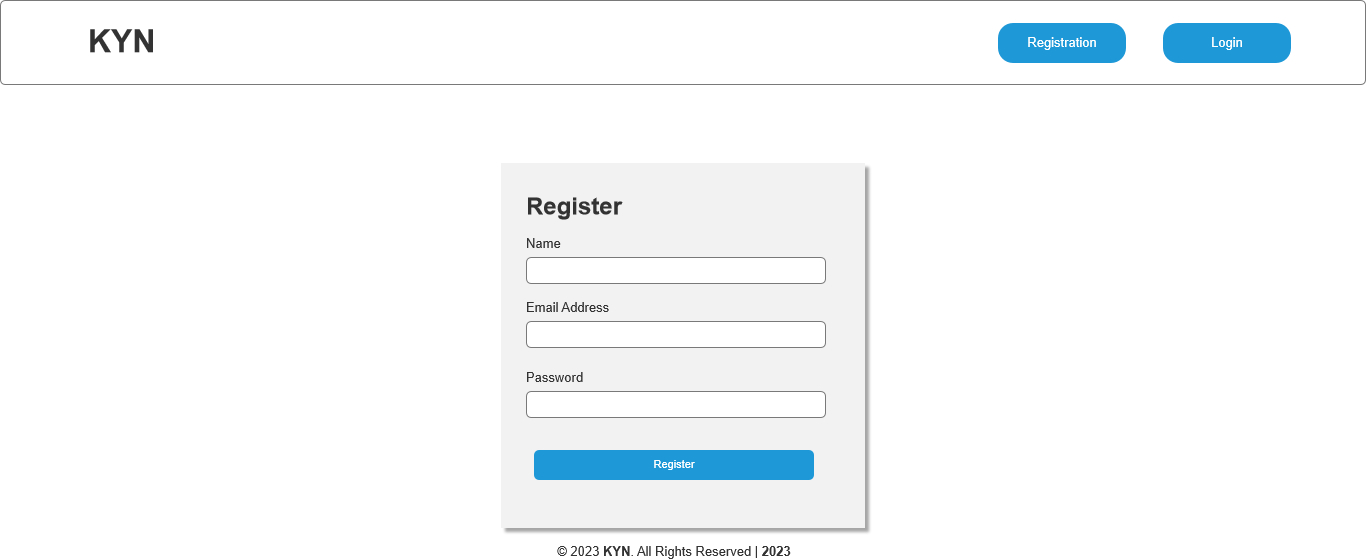
**Suitable Login API for KYN**

1. **Facebook Login API**

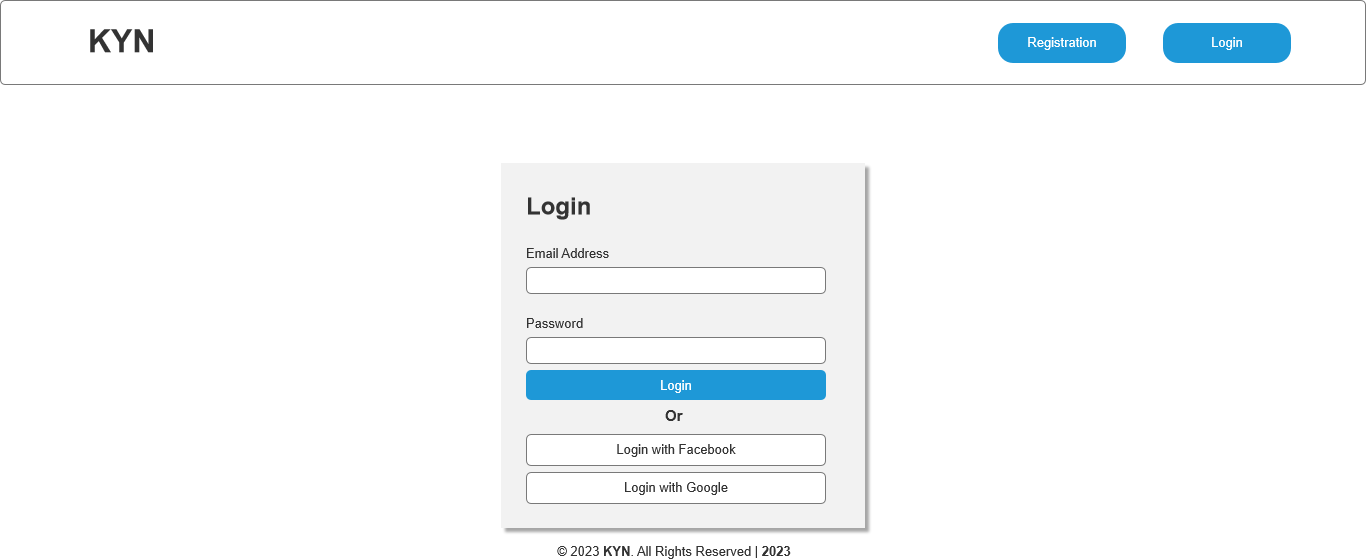
**Why Choose the API?**

As We explained, Facebook has more benefits than LinkedIn. Because Facebook was already popular in the early 2000s, many users were already using Facebook. Facebook is everywhere. Every device must have a Facebook account, meaning many facebook users exist. For Linkedin, not most people use the app only for professional, business purposes and others. That means easier reaching out to the users using Facebook.

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

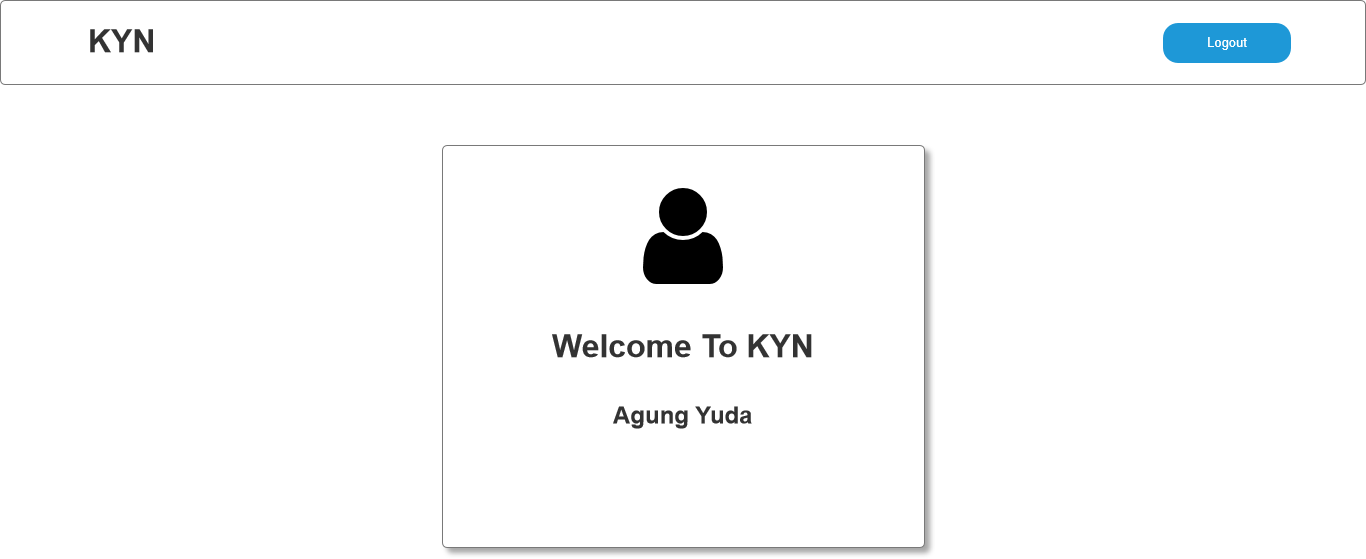


1. Login Page showing Google and Facebook Button and Form Input Login Wireframe



After user signing using google account with default scope, KYN website can access some user information like user’s google/facebook ID, name, and profile URL. With this information KYN website access the user to sign in to the website. After choose the account user will go to Dashboard page

1. Your KYN dashboard Wireframe after Google or Facebook login API



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

Google and Facebook's APIs use different languages ​​to build applications, so these two APIs are now available on most devices and run on most operating systems. For example, these APIs can be used in Windows and Mac environments as well as desktop and mobile devices.

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

****

The user needs to log in/register on the KYN website. First, a user clicks the login with Facebook button. then the user will be directed to facebook.com, where a login status check is done.

**Security of selected API for the application**

**Server-side authentication**

* **Basic Authentication Security:**

Basic Authentication Security: Basic Authentication is a widely used security mechanism that combines a username and password for authentication. It is a simple and straightforward method of authenticating API requests and protecting sensitive data. Basic Authentication operates over HTTPS, providing encryption of the credentials during transmission.

* **Justification:**

Simplicity: Basic Authentication is easy to implement and use, making it a practical choice for many applications.

Compatibility: It is supported by most programming languages and web frameworks, ensuring compatibility with various platforms.

* **OAuth**

OAuth is an open authorization protocol to give third-party applications 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 principle, 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.

**Client-side authentication**

Client-side authentication, the responsibility of storing and providing the authentication credentials lies with the client application. The client application securely stores the username and password and sends them in the request headers with each API call.

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

A well-liked and compact web application framework for Node.js is called Express.js. It offers a straightforward method for creating APIs and web applications. Express.js's simplicity and adaptability enable developers to effectively manage routes, middleware, and HTTP requests. It is appropriate for creating scalable, quick server-side applications.

* Spring Boot

A Java-based Spring Boot framework makes creating standalone, enterprise-level Spring apps easier. Its convention-over-configuration method enables developers to easily set up and configure the application with little boilerplate code. In addition to providing a complete environment, Spring Boot is renowned for its reliability, scalability, and broad community support.

* Laravel

The Model-View-Controller (MVC) architectural paradigm is used by the PHP-based web application framework Laravel. With features like routing, ORM, authentication, and caching, it is simple to construct online applications. It also offers an elegant and expressive syntax. Laravel is a well-liked option for PHP developers since it emphasizes developer productivity, maintainability, and simple code.

1. Three different of frontend

* React JS

A JavaScript package called React.js is used to create user interfaces. It allows developers to make reusable UI components and adheres to a component-based architecture. React.js effectively renders and updates components based on data changes, resulting in a fluid and engaging user experience. Mobile and single-page applications (SPAs) are frequently built using it.

* Angular JS

Google maintains the feature-rich JavaScript framework known as Angular.js. It offers a fully functional development environment for creating web apps. To build scalable and maintainable systems, Angular.js uses declarative templates, two-way data binding, and dependency injection. It can provide comprehensive testing and features like routing and form validation.

* Vue JS

A progressive JavaScript framework for creating user interfaces is called vue.js. It is renowned for being straightforward, adaptable, and simple to integrate into current projects. Vue.js is appropriate for small and large-scale projects since it allows developers to absorb its features gradually. It features a component-based architecture and a reactive data-binding framework.

1. Three different API

* REST API

A RESTful API interface allows two computer systems to communicate safely over the internet. To complete various activities, most business apps must interface with other internal and external applications. For instance, to automate invoicing and connect with the internal time attendance application, your internal account system must exchange information with your customers' banking system to generate monthly payslips. RESTful APIs support this information exchange because they adhere to safe, dependable, and effective standards for software communication.

* SOAP API

SOAP, or Simple Object Access Protocol, is a standard web communications protocol developed by Microsoft in 1998 and is an important part of Service Oriented Architecture (SOA). SOAP is used to expose web services and can integrate various communication protocols using the HTTP protocol. SOAP uses Extensible Markup Language (XML) to send messages and is a complete and official protocol with built-in security features and strict usage rules. While SOAP has clear advantages of security and standards, it is also a complex protocol and requires more bandwidth than some alternatives.

* RPC API

Remote Procedure Call (RPC) builds distributed client-server-based systems. It is also known as a function call or a subroutine call. The called process does not have to be present in the exact same address space as the calling procedure because it is based on traditional local procedure calling. RPC is ideally suited in a client-server interaction where the control flow lingers between the two. The execution thread switches between the client and server rather than running on both simultaneously.

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

Front-end development environments:

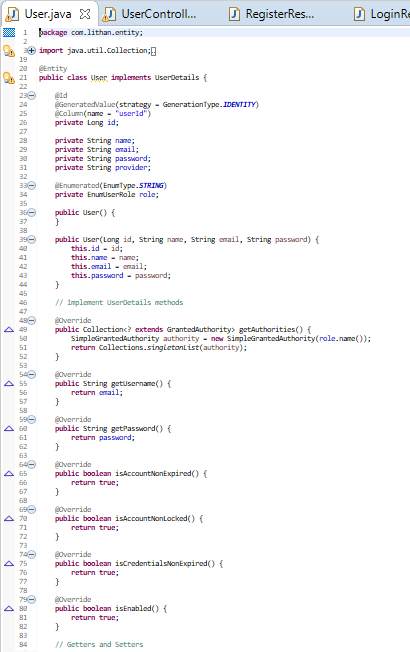
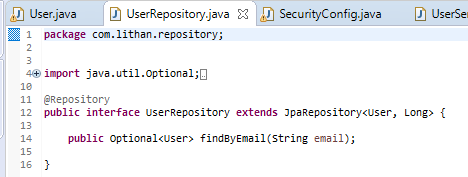
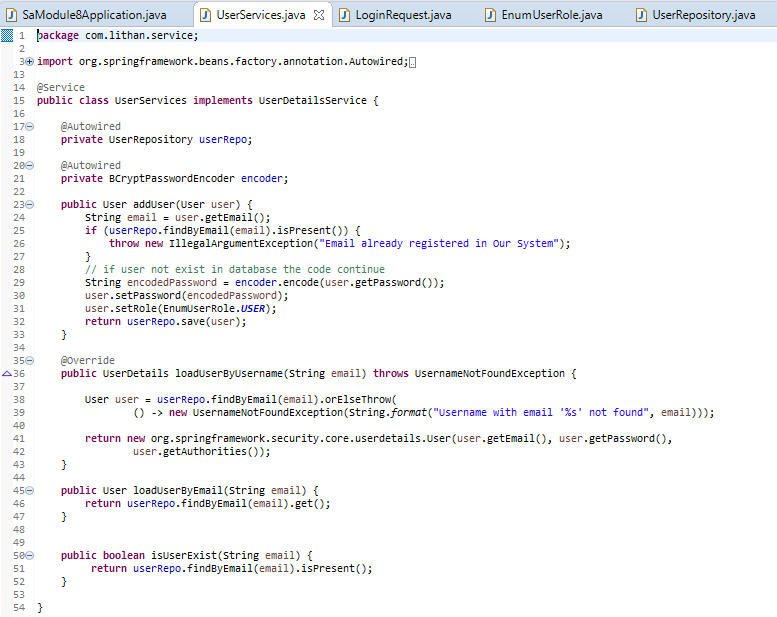
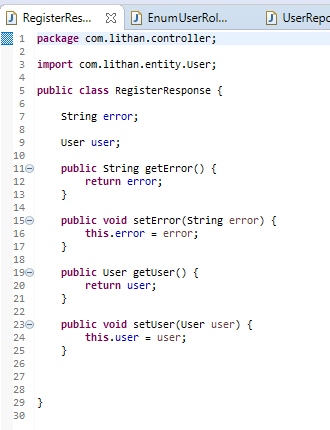
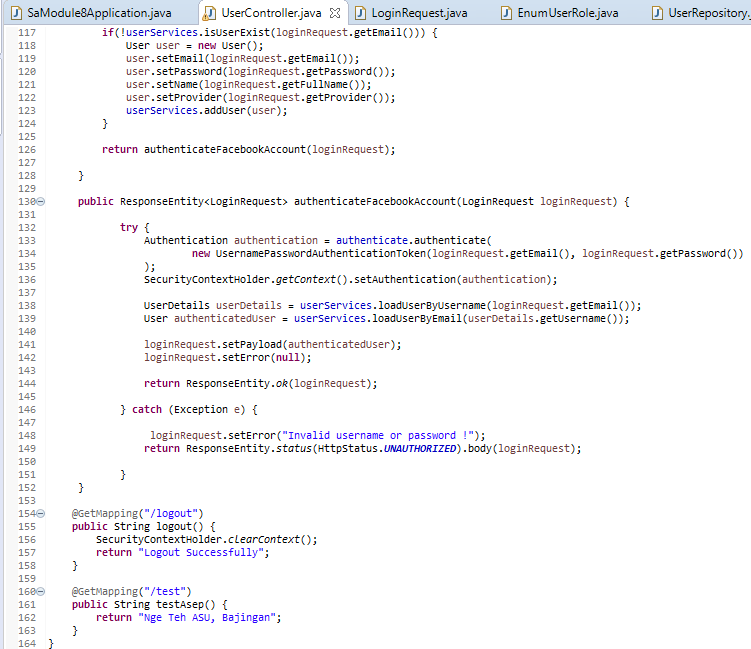
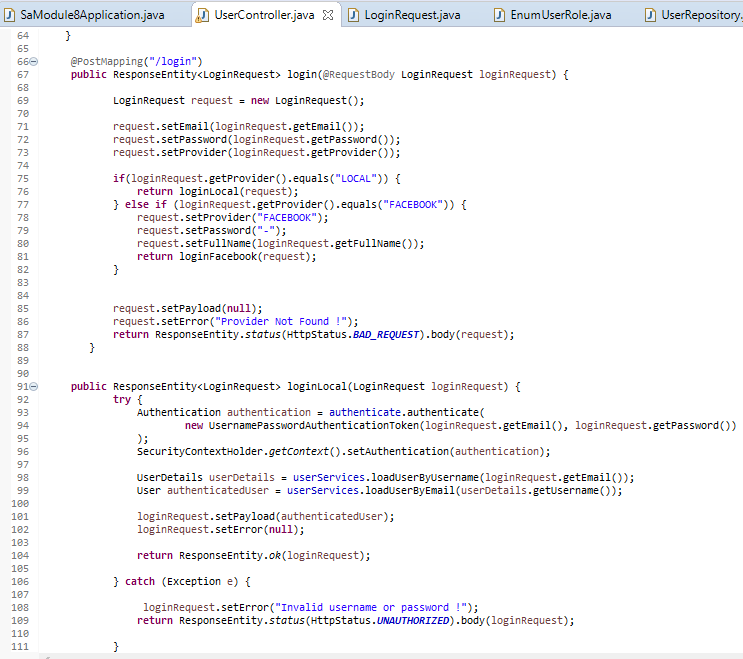
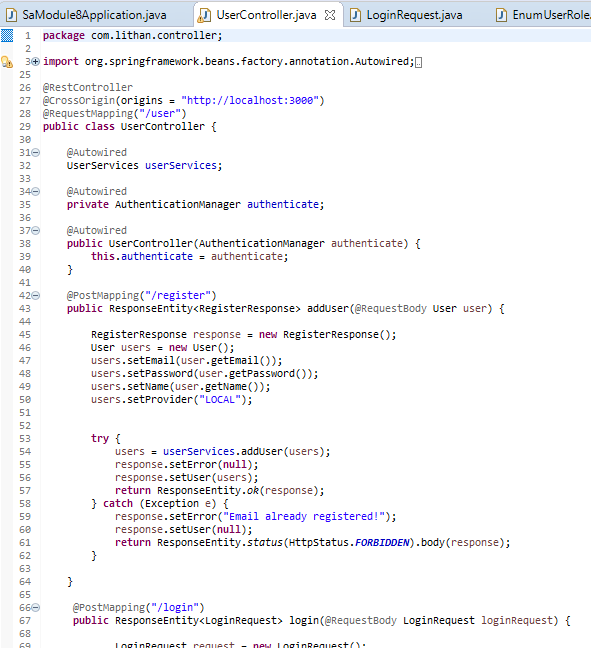
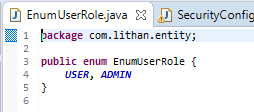
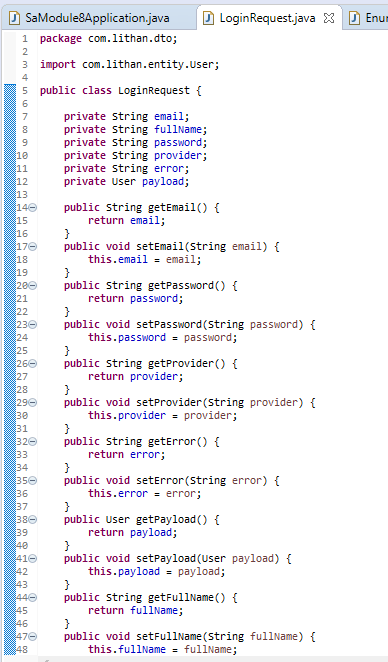
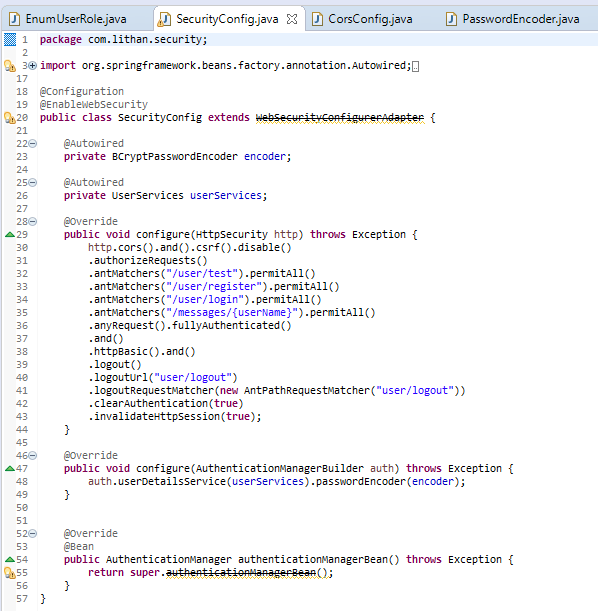
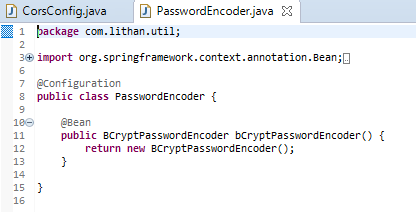
* Visual Studio Code: a popular open-source code editor with built-in debugging that supports Git and several programming languages.
* Sublime Text: a powerful, portable text editor with an easy-to-use interface, a wide selection of plugins, and customizability possibilities.
* WebStorm: an effective integrated development environment (IDE) for front-end languages like TypeScript and JavaScript.

Back-end development environments:

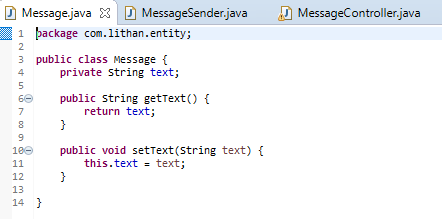
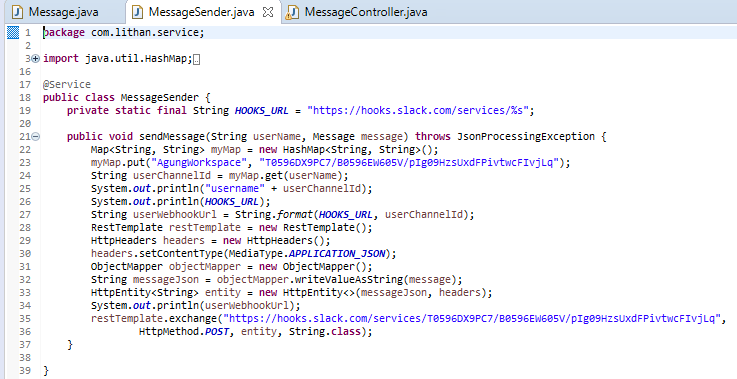
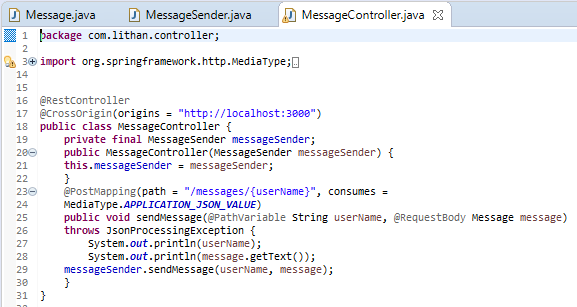
* Visual Studio: Microsoft developed a well-known IDE, packed with features, and supports a wide range of programming languages and frameworks.
* Eclipse: is an open-source, widely used IDE supporting numerous programming languages and frameworks.
* IntelliJ IDEA: an established integrated development environment (IDE) for Java development that fully features and supports a wide range of additional programming languages and frameworks.

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

**Screen Capture of KYN Website Back-End**

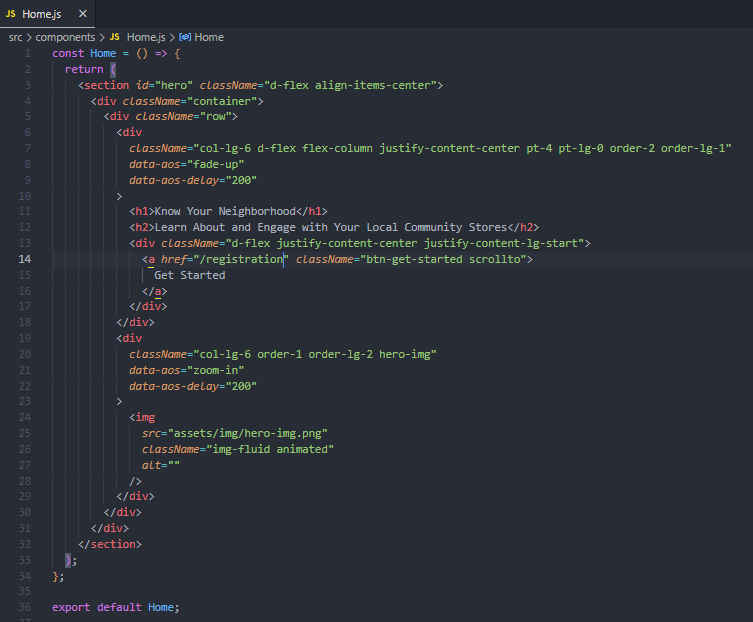
1. **User.java**
2. **UserRepository.java**
3. **UserServices.java**
4. **RegisterResponse.java**
5. **UserController.java**
6. **EnumUserRole.java**
7. **LoginRequest.java**
8. **SecurityConfig.java**
9. **CorsConfig.java**
10. **PasswordEncoder.java**

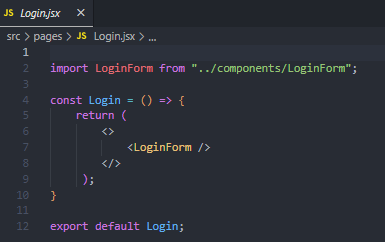
**Message**

* + - 1. **Message.java**
      2. **MessageSender.java**
      3. **MessageController**

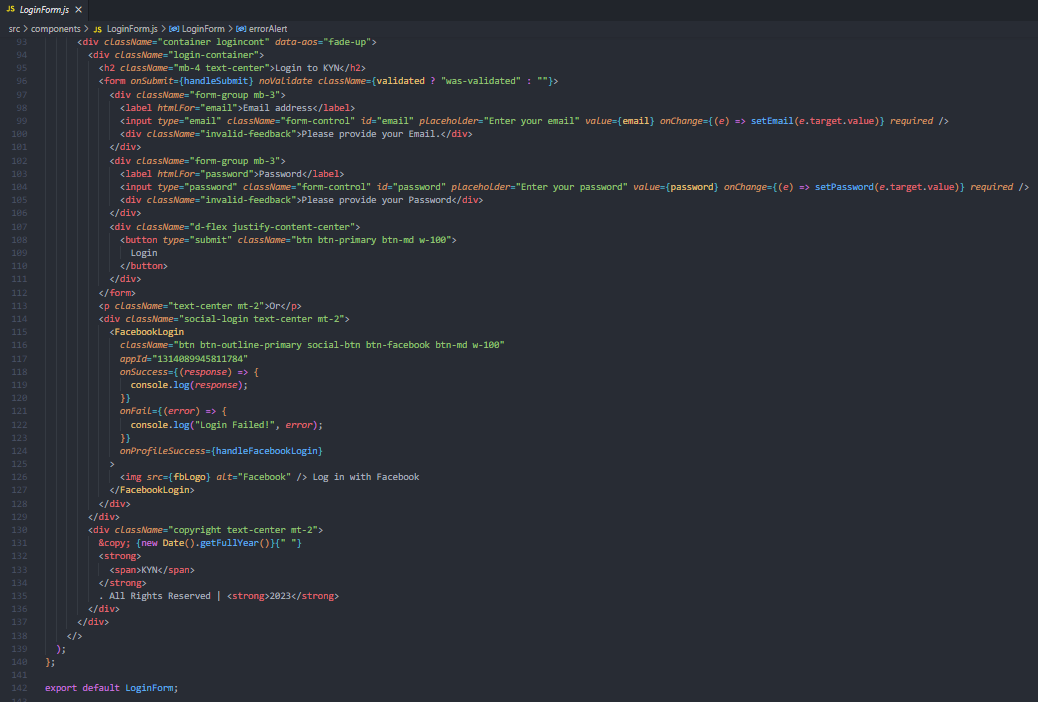
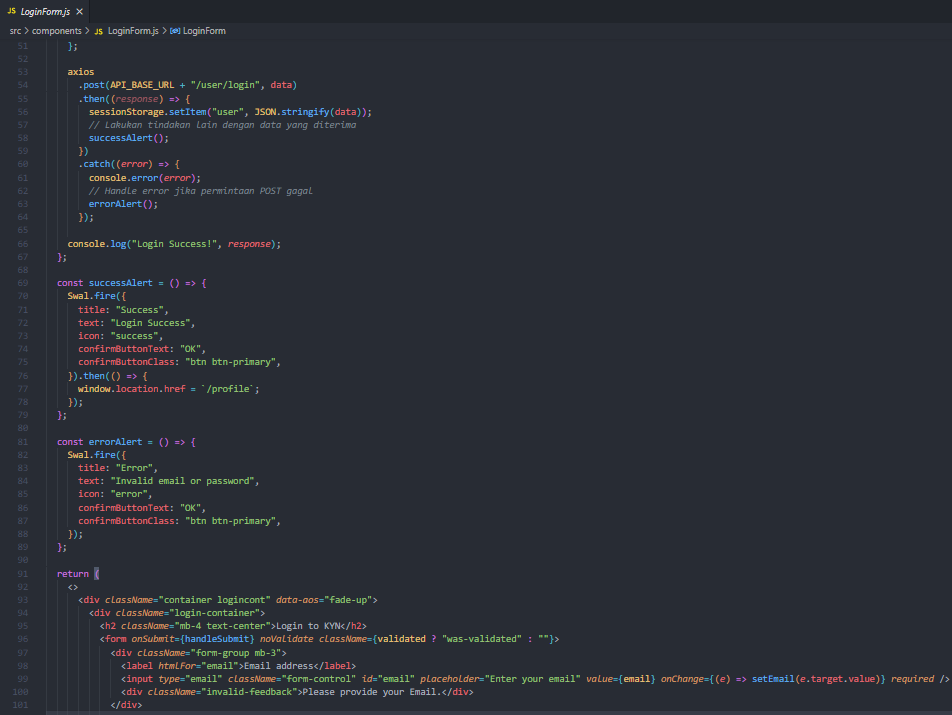
1. **Develop an application that utilizes an API.**
2. **Develop front-end using react**

* **Home.js**



* **Login.jsx**

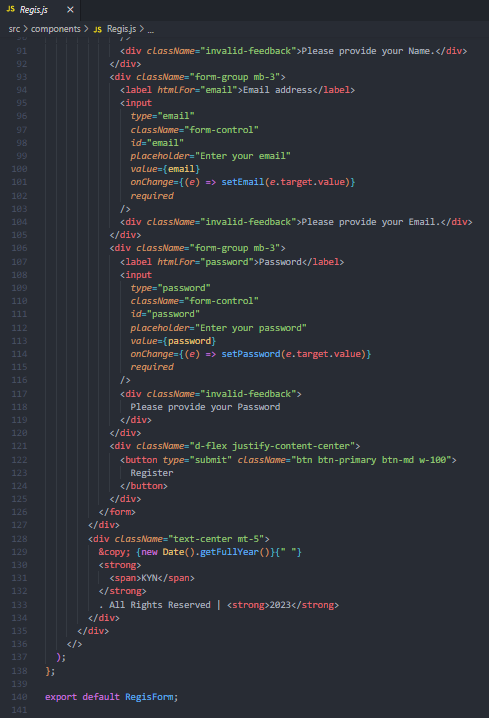
**LoginForm.js**



* **Registration.jsx**

**Regis.js**





**Message in Contact Us  
Contact.js**



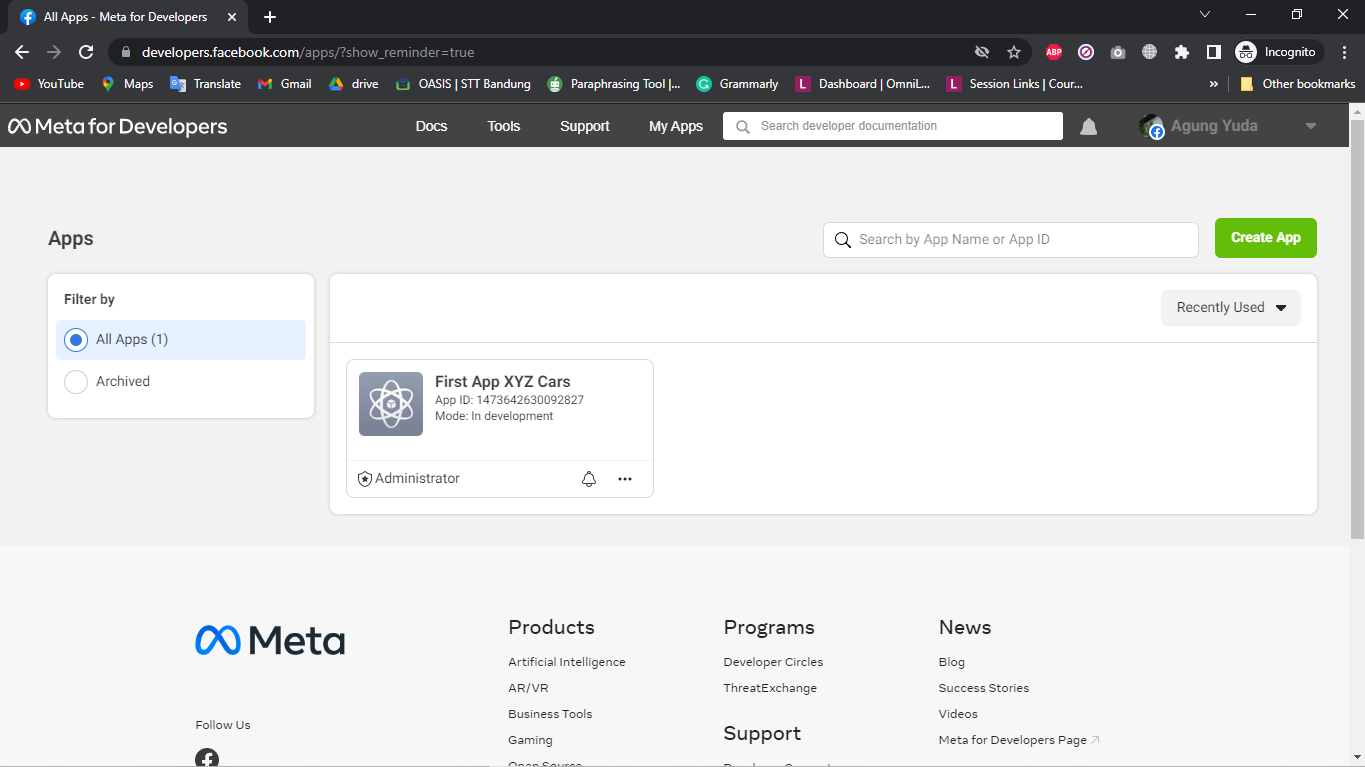
**Service.js**

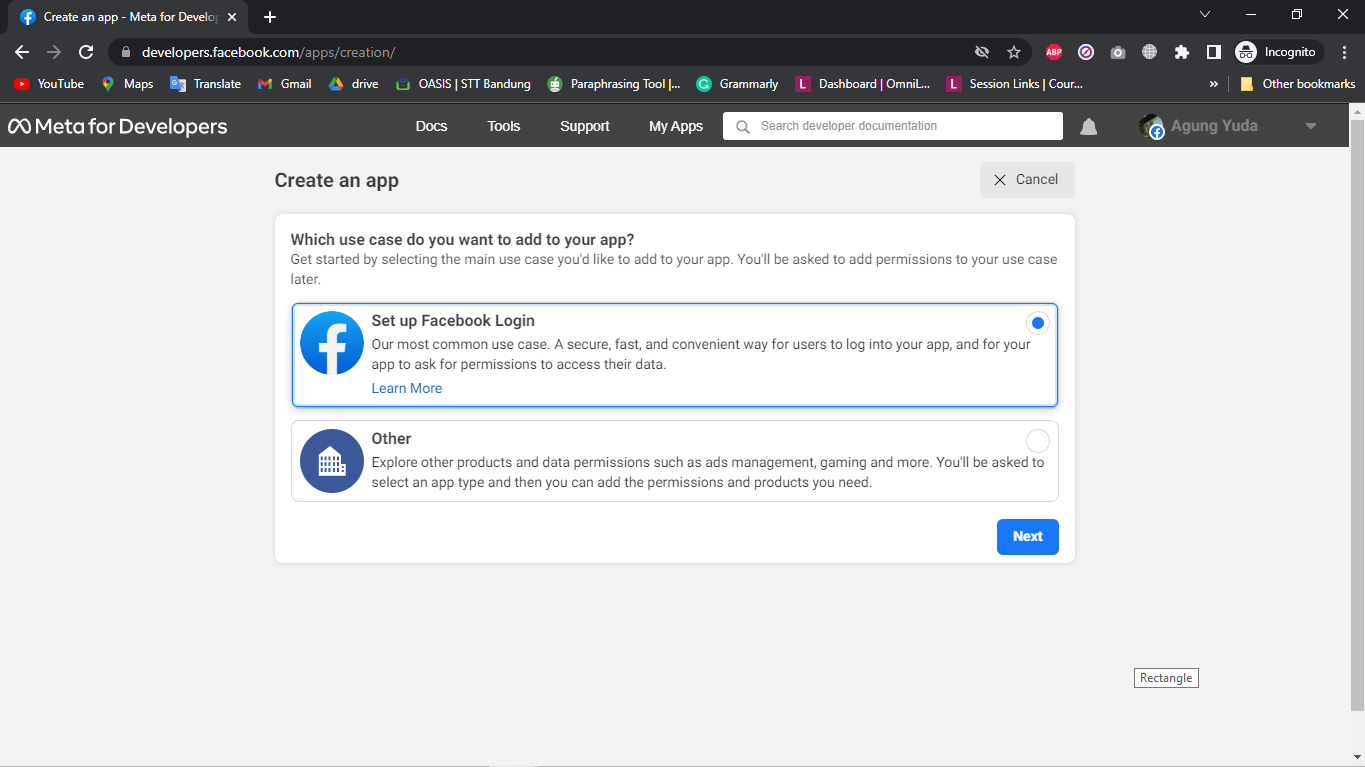
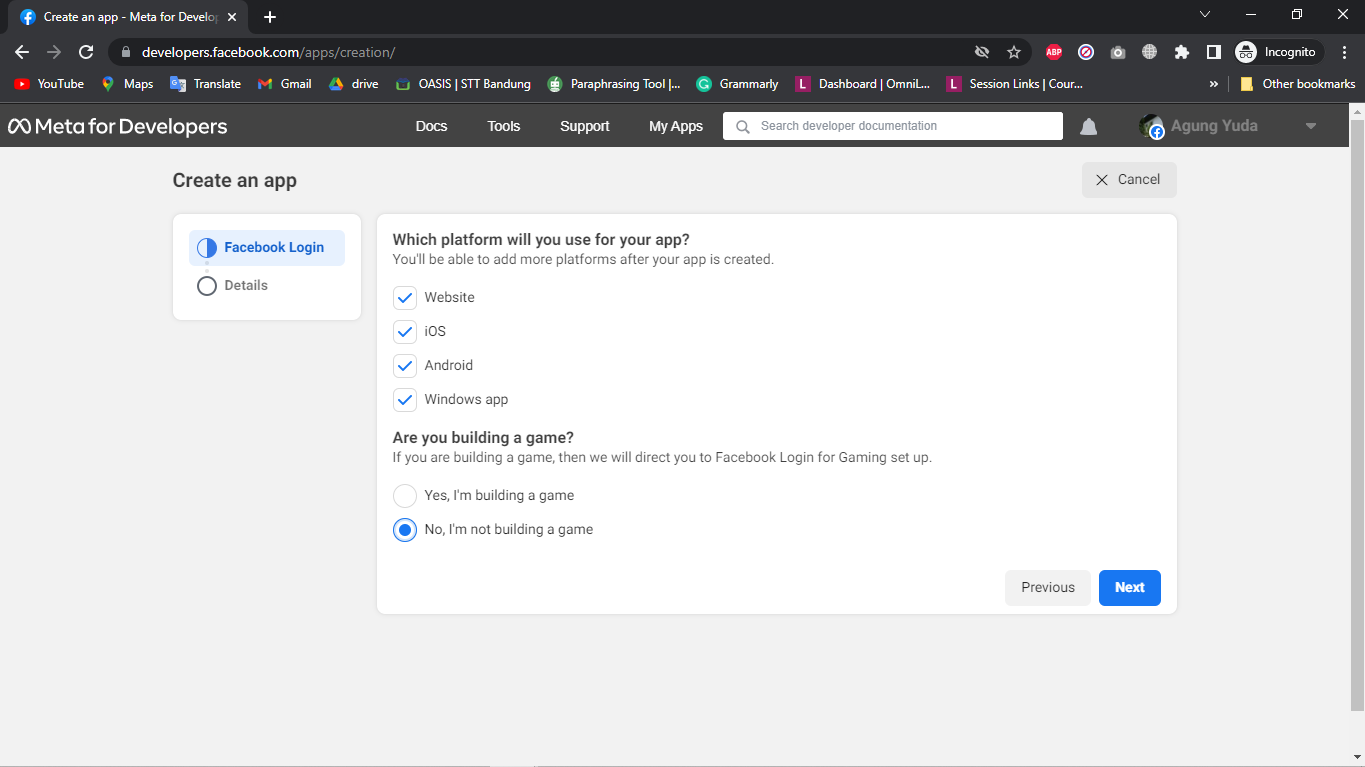
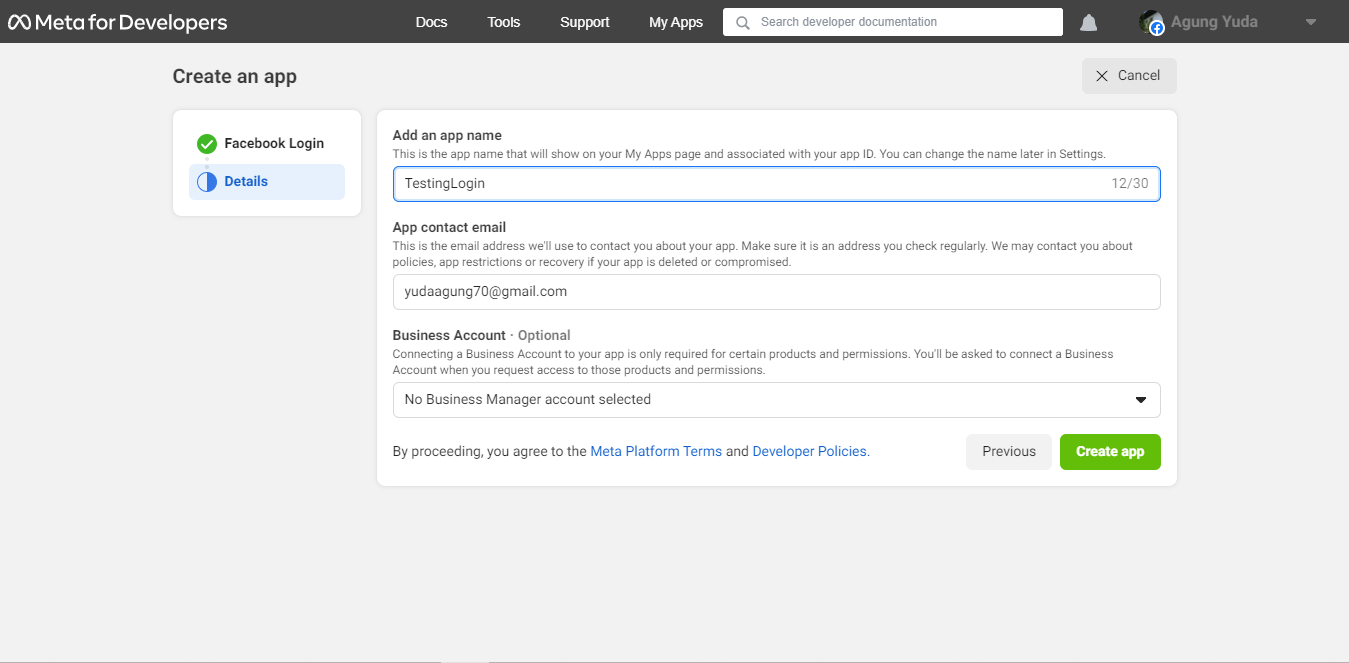
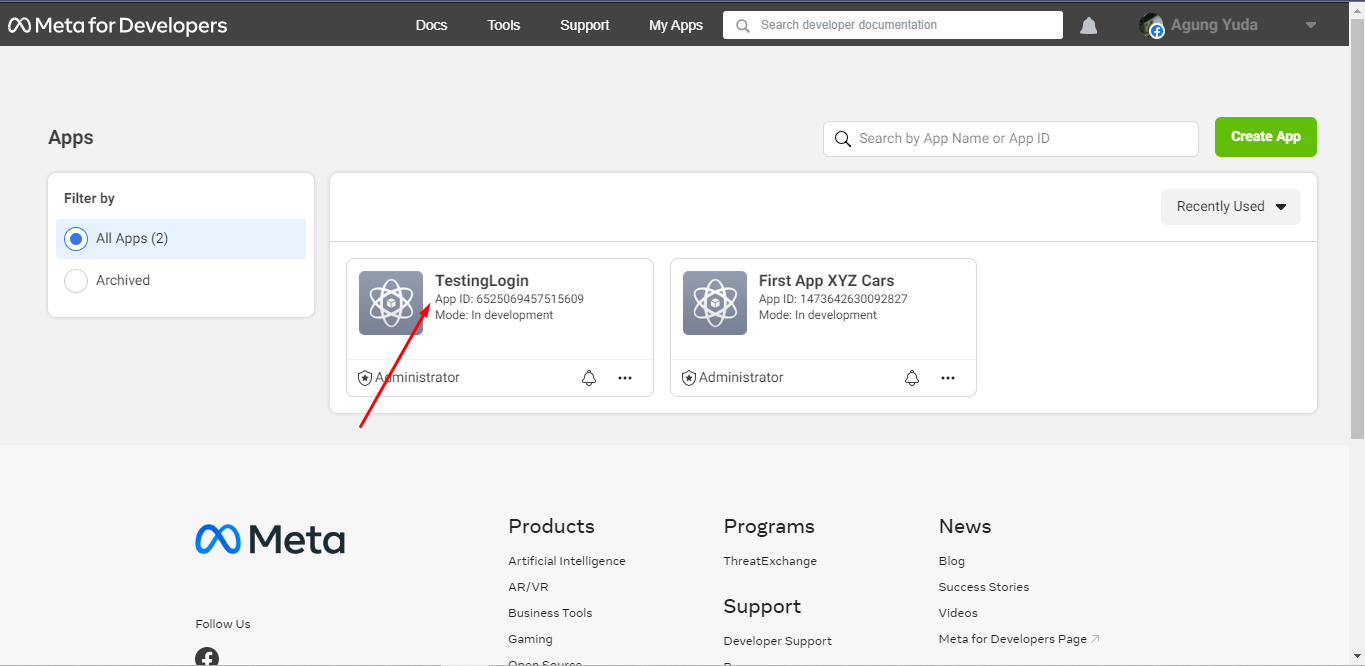
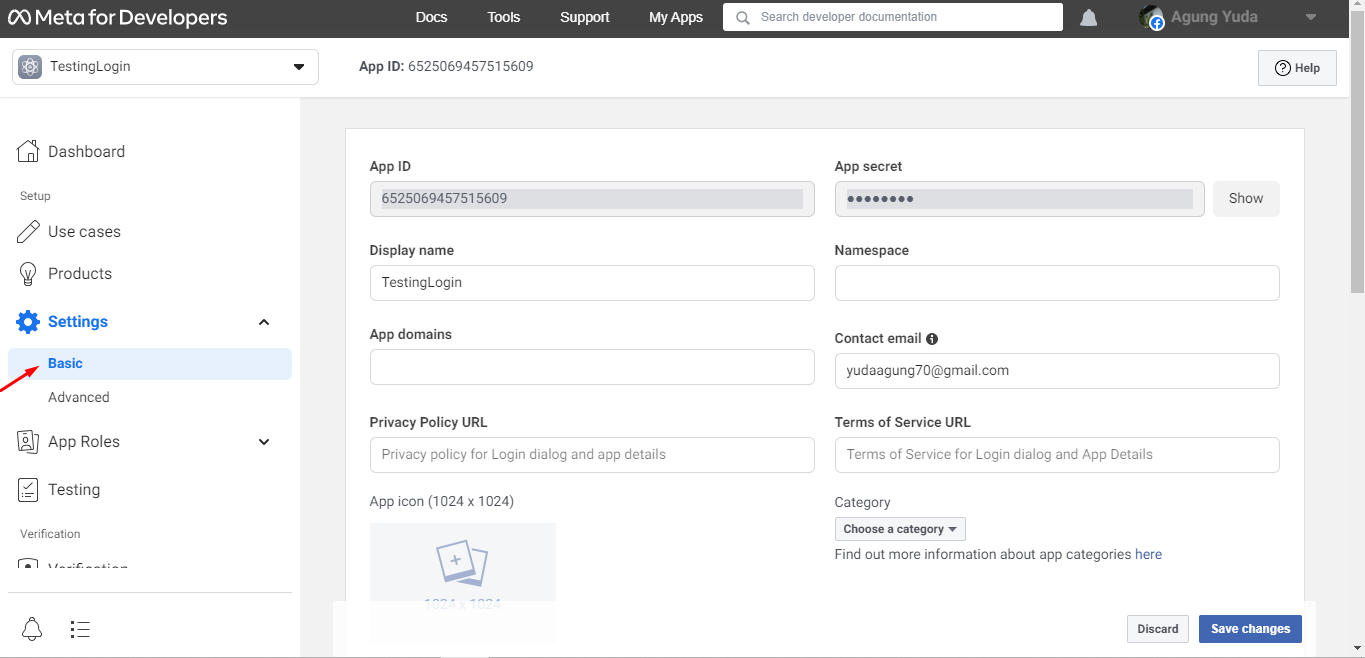


1. **Construct the application which implements the selected API in Task 2.**
2. **How to Login/Register with Facebook API**
   * Firstly, open meta for developer in google then login into your account



* + create an app by clicking “create app” below



* + Choose Facebook Login  
    
  + Checklist by the needs  
    
  + Name the app  
    
  + Open Your App  
    
  + Go to setting and basic, and you can see your APP Id  
    
  + Then copy and paste “App ID” and to the code editor in your facebook login file  
    

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 is done to test the software by analyzing and examining the internal structure and code. Unlike black box testing, which only looks at the input and output results of the software, white box testing focuses on the input and output flow of the software.

* Why are we using white box testing?  
  Advantages
  + 1. Improve accuracy in implementing software.
    2. Makes it easier to find errors or bugs in software that were not previously visible.
    3. It Makes testing easier because it is done thoroughly to minimize the possibility of errors in the code.
    4. Minimize errors or bugs because testing can be done before the software launches.

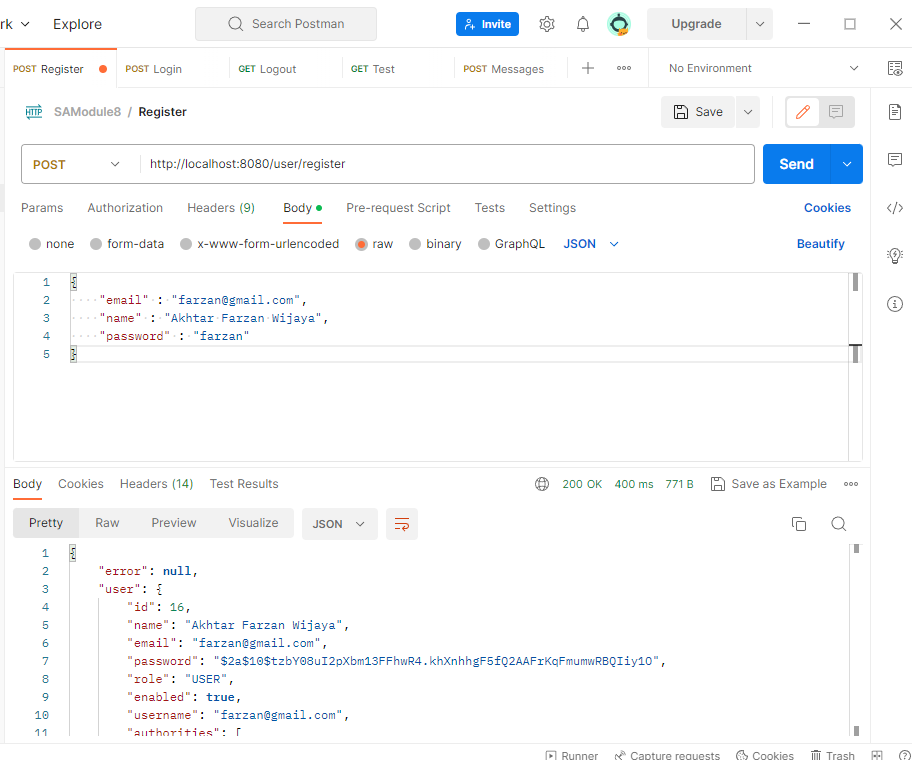
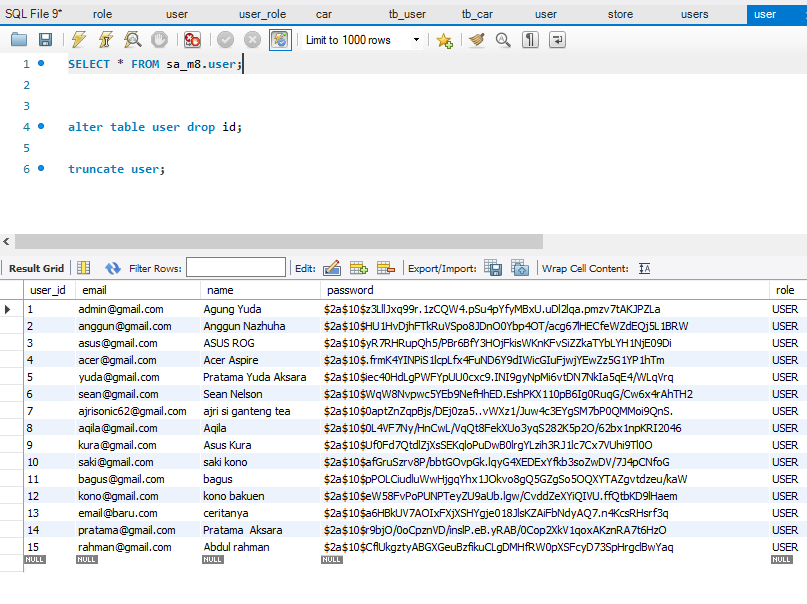
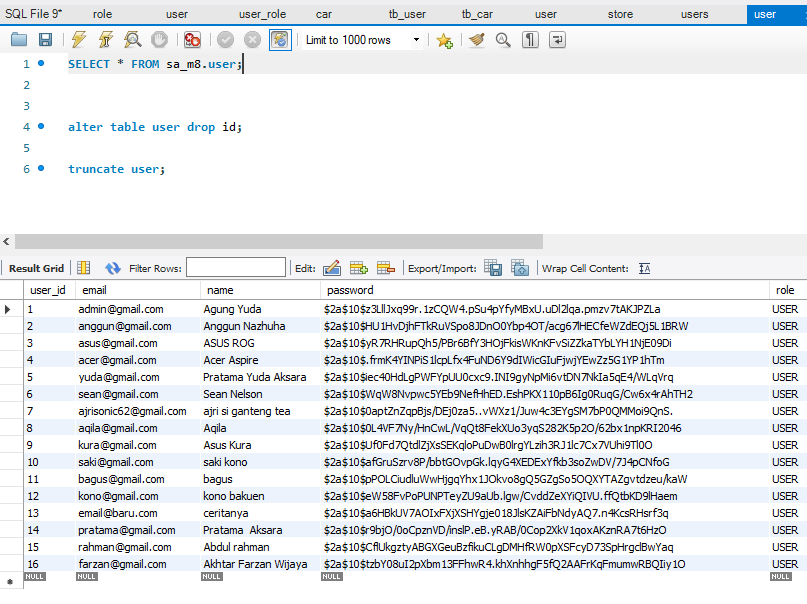
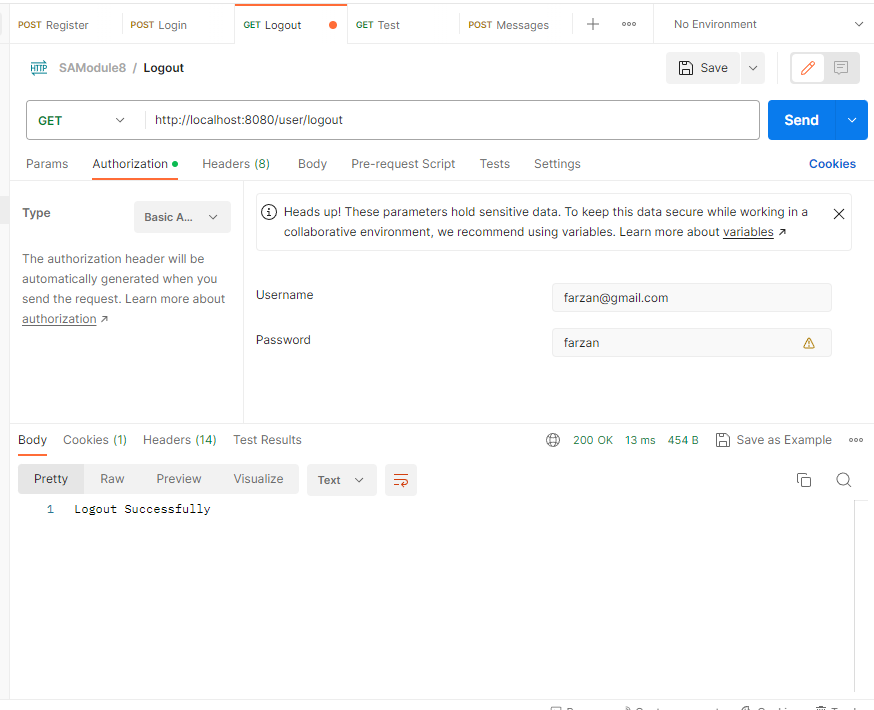
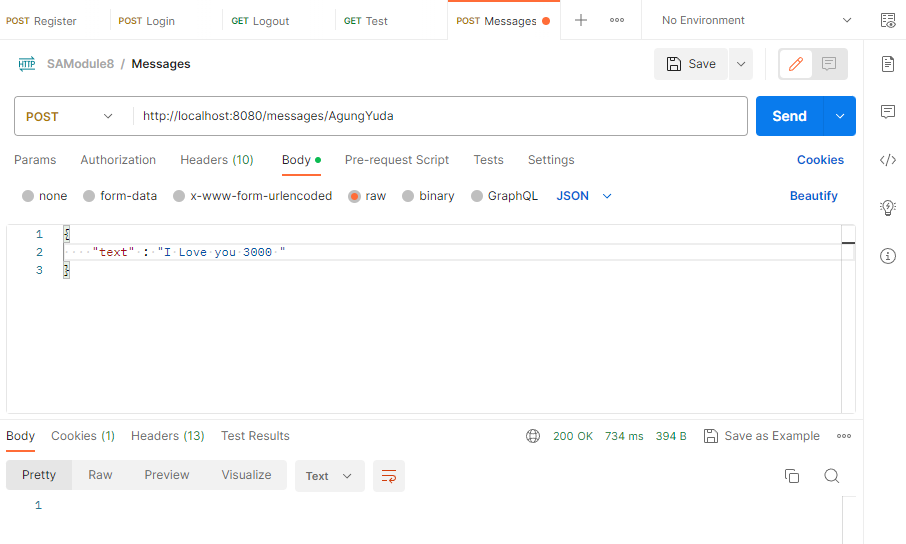
White box testing is used to:

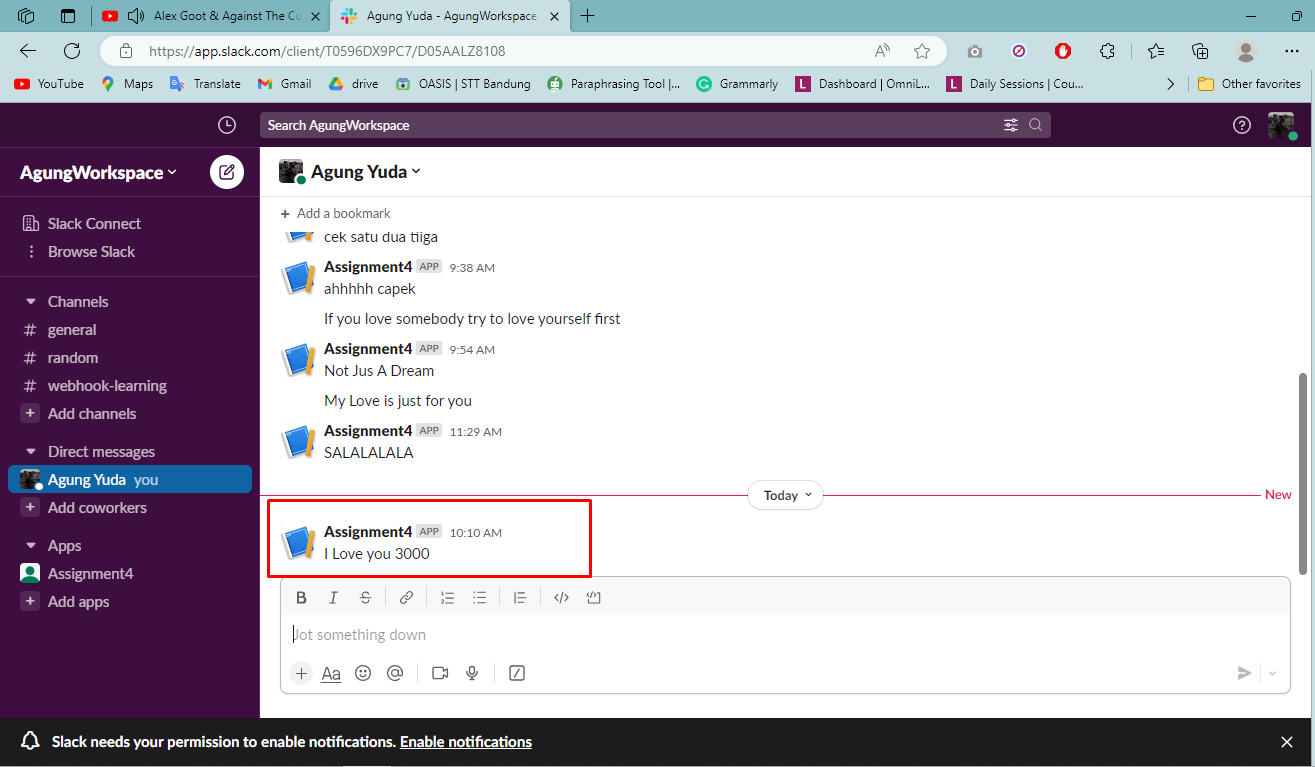
• Ensure that each independent path within a module is executed at least once.

• 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.

* Test all controller mappings using POSTMAN tool and then record your result
* Register  
    
  Database Before Register  
    
  Database After Register  
  
* Login  
  
* Logout  
  
* Message  
  



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

* What is black box testing?

Black box testing, also known as Behavioral Testing, is a test carried out to observe the input and output results of the software without knowing the code structure of the software. This test is carried out at the end of making the software to find out whether the software can function properly.

* Benefit black box testing?
  + - 1. Testers do not have to know a programming language.
      2. Testing is carried out based on the user's point of view. This is done to find inconsistencies in the software.
      3. Developers and testers depend on each other.
      4. Testers don't need to check the code.
      5. Allows testers and developers to work independently without interfering with each other's work processes.
* UAT testing

|  |  |
| --- | --- |
| **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 Data Table

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Data** |
| TS001 | TC001 | Name = Saka Raya Bumi  Email = saka@email.com  Password = (not inputed) |
| TC002 | Name = Saka Raya Bumi  Email = saka@email.com  Password = saka321 |

* Test Evidence

|  |  |  |
| --- | --- | --- |
| **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 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 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.**

* The application development process
* Strengths and Weaknesses selected API, features
* Strengths REST API

1. It can be used by various programming languages, including the various platforms it uses.
2. Simpler, especially when compared to using SOAP
3. Easier to learn
4. Like the web, which always uses HTTP in every part of it.
5. Android applications that use the Rest API are much faster than web view-based Android applications.

* Weakness REST API

1. Usually longer access times compared to native libraries
2. More vulnerable to security attacks because they have to pass the HTTP protocol
3. **Provide data security report of your developed application.**

In applications that have been developed, data security is very important to maintain the confidentiality and integrity of user information. Therefore, we implemented Spring Security as a security mechanism in the backend of KYN applications.

Our Spring Security configuration defines several steps to ensure effective security. First, we disabled the CSRF feature to avoid potentially damaging Cross-Site Request Forgery attacks. In addition, we allow access to certain URLs without requiring authentication, such as "/user/test", "/user/register", "/user/login" and "/messages/{userName}". This allows users to perform actions such as registration, logging in, and accessing messages without having to authenticate first.

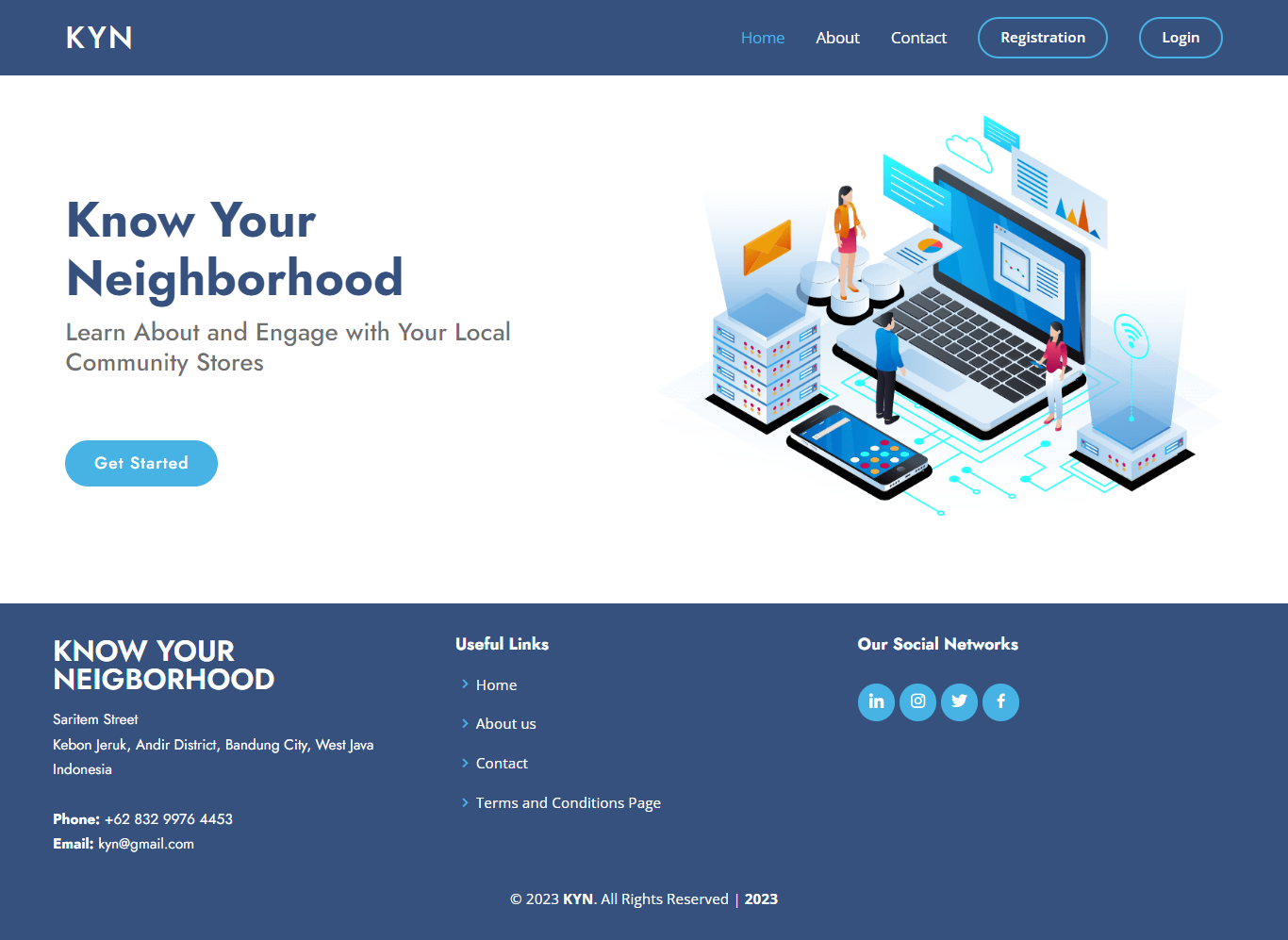
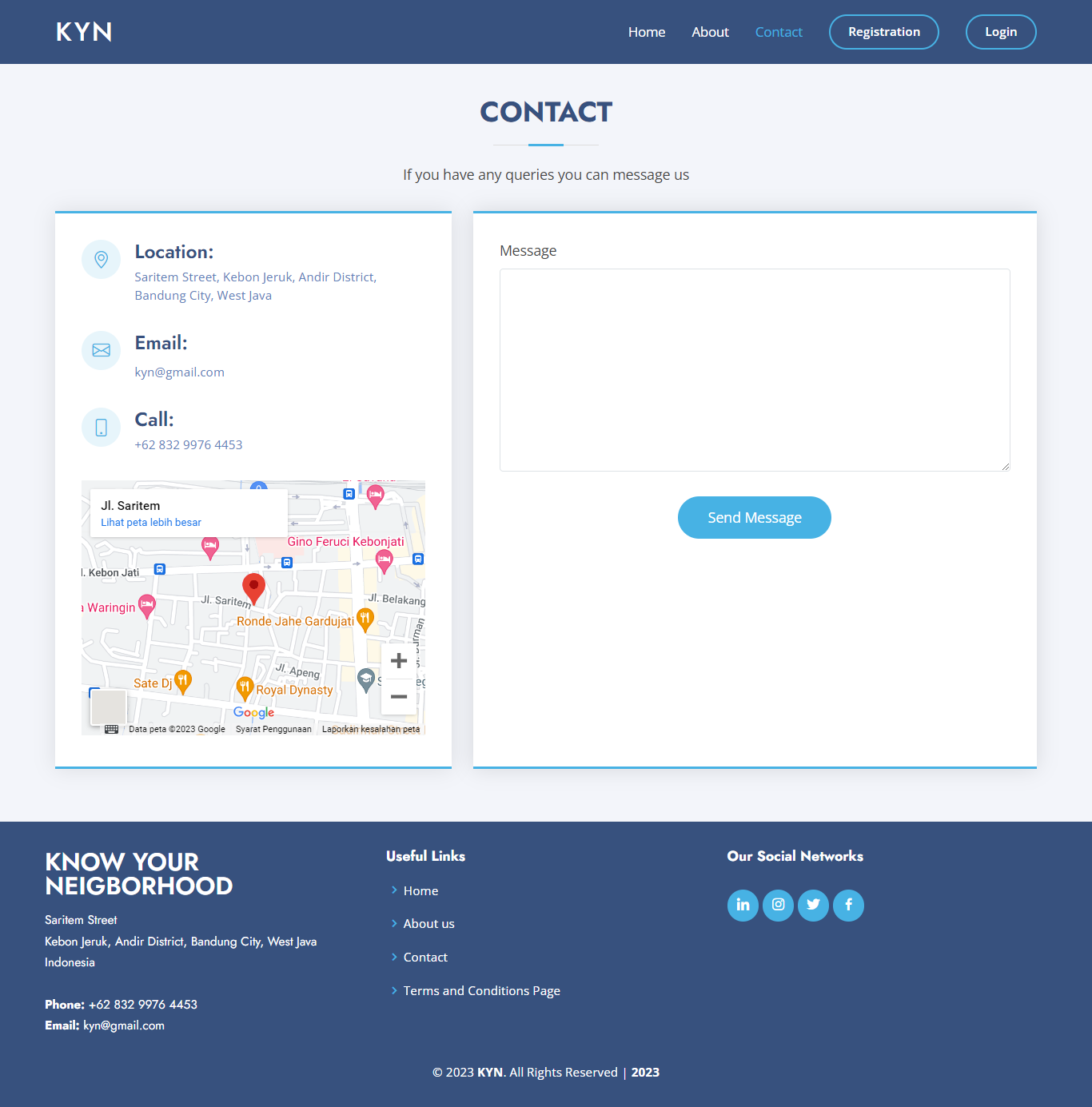
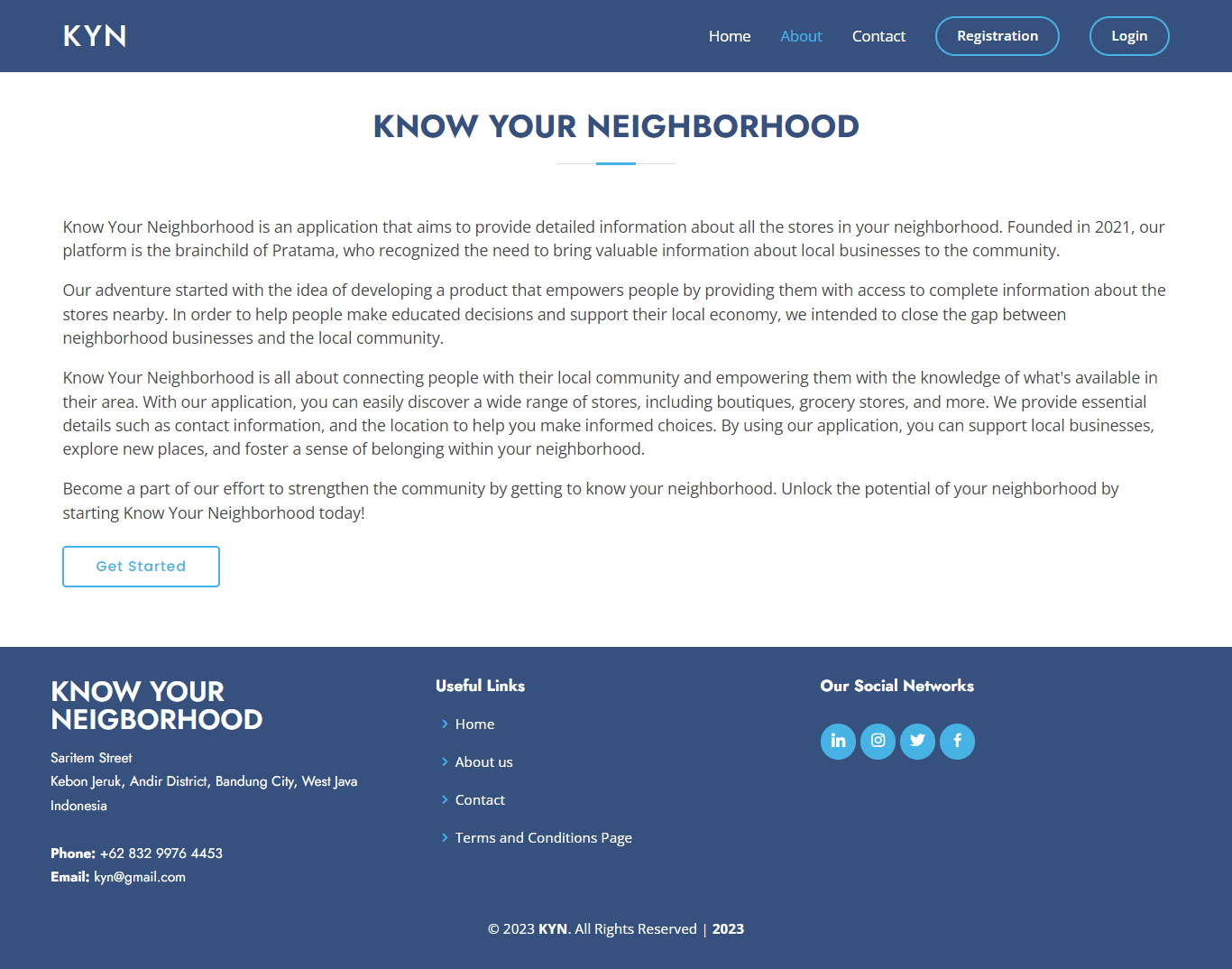
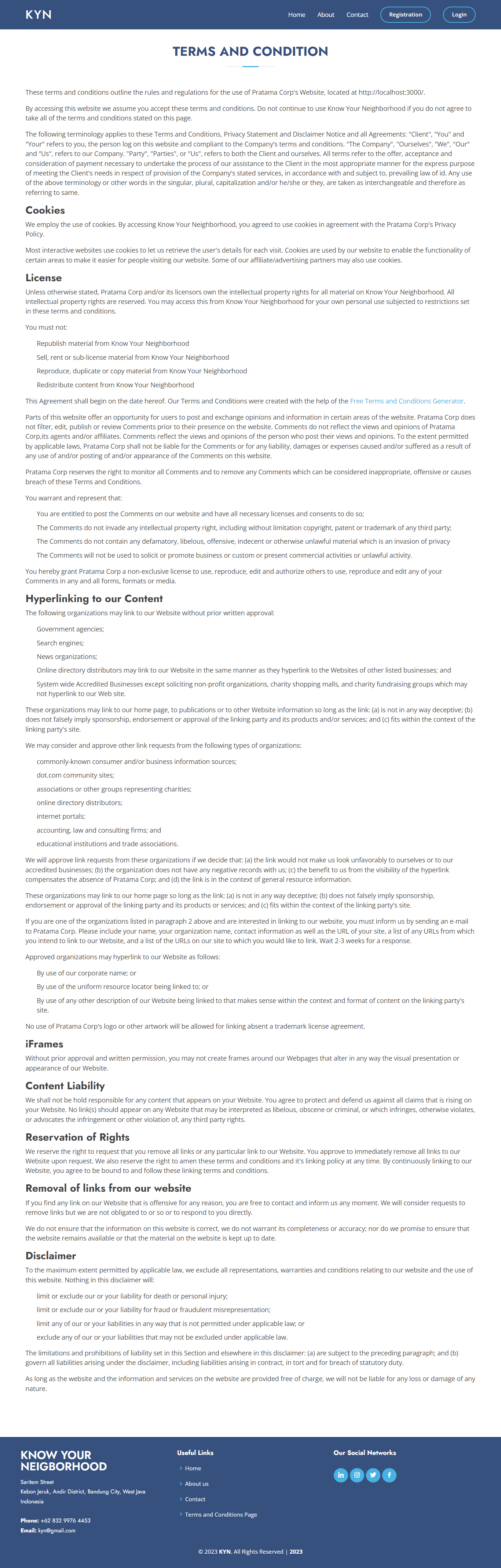
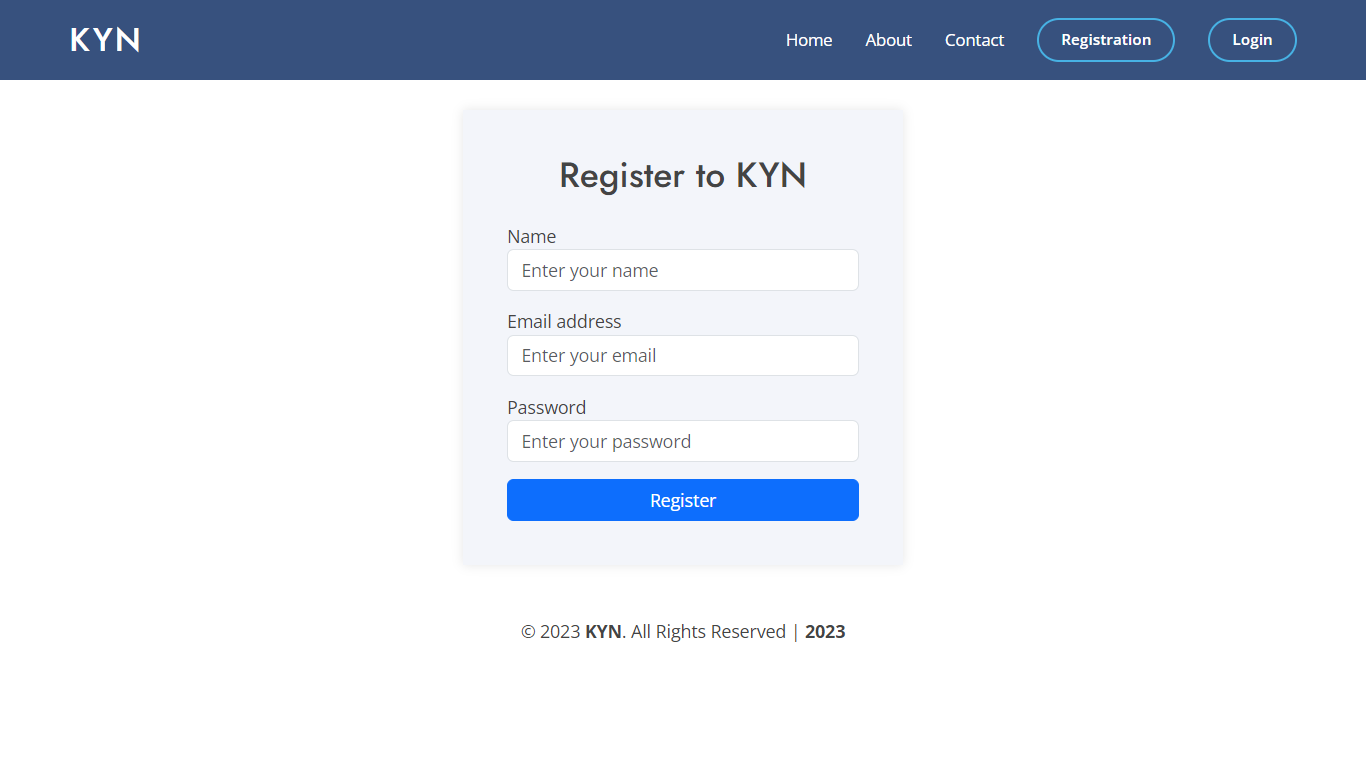
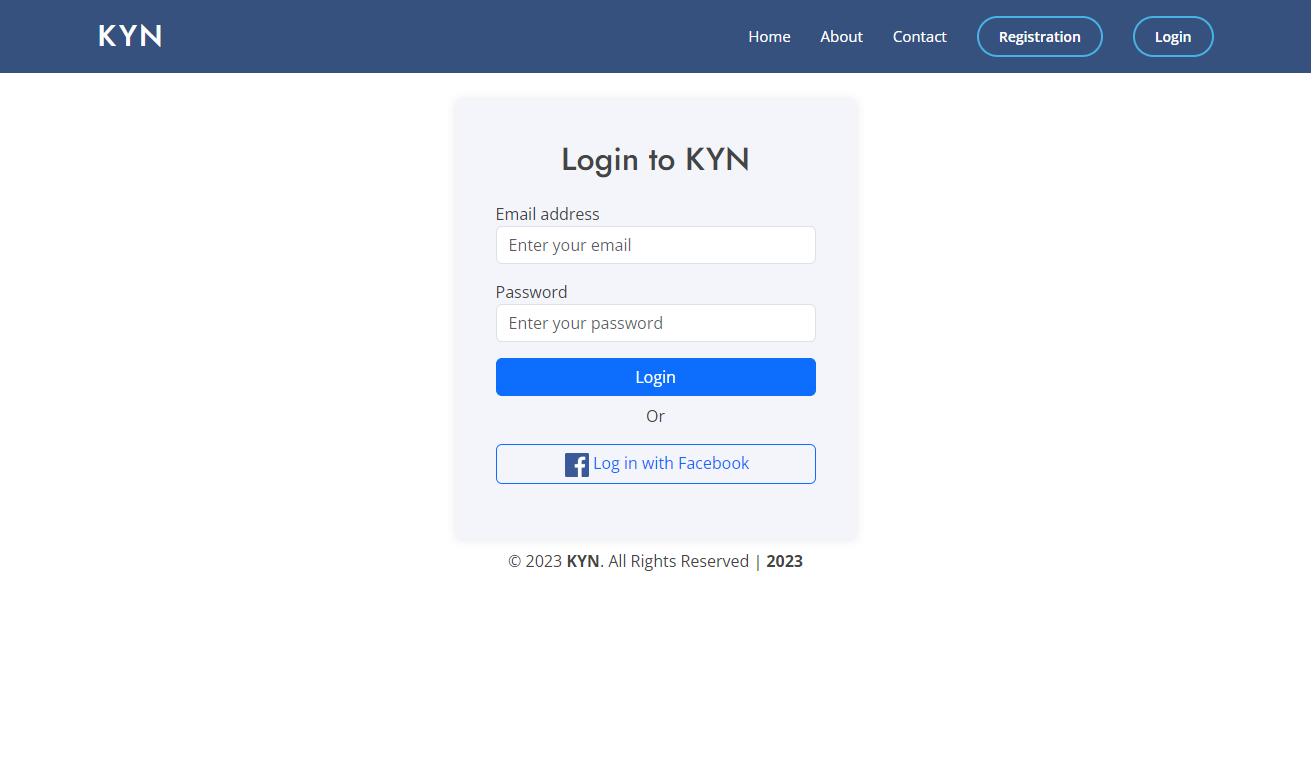
However, to protect more sensitive data, we require users to authenticate all other URLs fully. We use basic HTTP authentication to ensure reliable authentication security. Additionally, we have configured a URL for the logout process, so users can safely exit the app and have their authentication session removed.

To secure user passwords, we use BCryptPasswordEncoder provided by Spring Security. This allows us to encrypt passwords before storing them in the database, maintaining confidentiality and preventing unauthorized access.

Additionally, we've implemented a UserDetailsService used by Spring Security to retrieve user information from the database. With this, we can manage and verify user authentication information efficiently and securely.

In addition to backend security, we manage CORS (Cross-Origin Resource Sharing) policies using the CorsConfig configuration. We restrict access only from a predefined origin, i.e., "http://localhost:3000". Also, we only allow certain HTTP methods, such as GET, POST, PUT, and DELETE. This helps prevent potentially damaging cross-domain attacks and maintains data integrity.

By implementing Spring Security, Authentication and proper CORS configuration, we have ensured that KYN applications are secure. The intelligent and scalable authentication and authorization features provided by Spring Security help protect user data effectively and provide a secure user experience.

1. Task 6 Screen Capture
   1. Home Page  
      
   2. Contact Page  
      
   3. About Page  
      
   4. Terms and Condition Page  
      
   5. Register Page  
      
   6. Login Page  
      
   7. Profile Page  
      