**ICIN Bank**

A dynamic and responsive Java online banking web application to deposit, withdraw, and transfer the money between the accounts.

## Developer details :

Name = Neeti Kulkarni

## Sprints planning and Task completion:

### For this project, I’ve planned to complete the project in 4 Sprints. And the task is completed according to that. And this is a Full Stack Java Project so Backend & Frontend Technologies used here

### 1st Sprint:

• Creating the flow of the application. • Initializing the git repository to track changes as development progresses. • Creating a Spring boot application to fulfill user requirements. • Configure the Database to maintain the data used for the application. • Adding the Required dependencies used for the application.

### 2nd Sprint:

• Implemented the Model Layer and Service Layer for the project. • Implemented the Business Logic part. • Creating API methods to fetch the response. • configure the API to check the Response of the requested object. • Testing in the Post-Man Tool to check whether the API methods are working or not by Response.

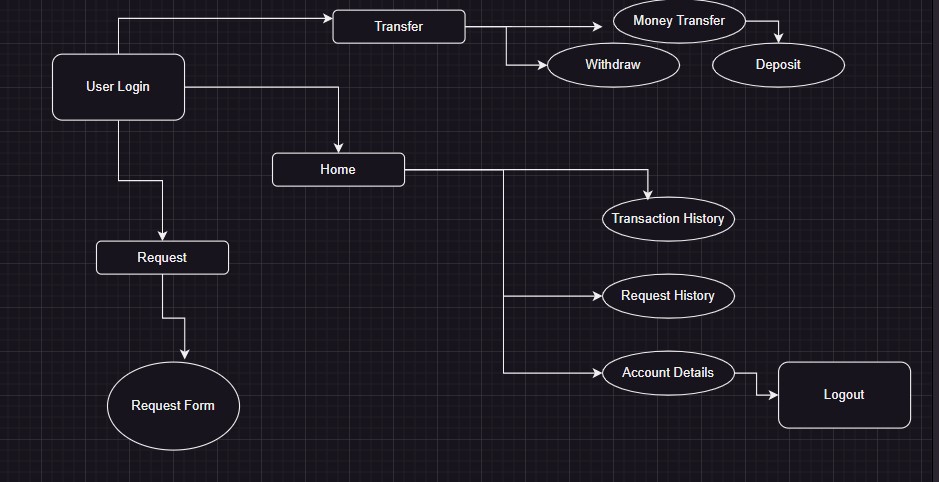
### 3rd Sprint:

• Creating an Angular application to fulfill user requirements. • Adding the Required Packages used for the application. • Creating Components and Services, Styling the Pages. • Creating Reactive Forms and Validations to the Forms

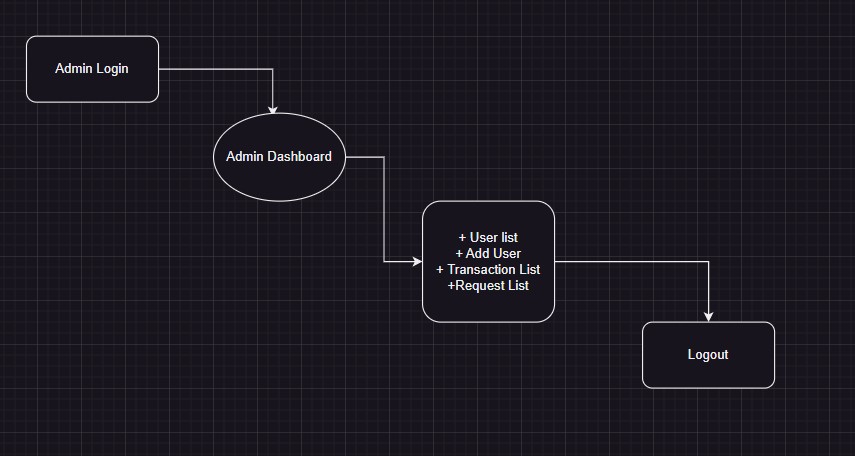
### 4th Sprint:

. • Creating Service methods to fetch the response. • Passing the Response from the Service to the component class. • Implemented the Query params using routing. • Integrated the Payment Gateway using Stripe API for payments.

## User Flow of Project:



## Admin Flow of Project:



## Core concepts used in the project:

Concepts used in Project: • HTML, CSS & JS • ANGULAR • JAVA , API, MAVEN • SQL DATABASE • SPRING TOOL SUITE 4 • Tomcat Server 9.0 • Git & Github • POSTMAN • VS Code

• Object-Oriented: used to create and model objects for users and their credentials.

• Reactive Forms: Reactive forms provide a model-driven approach to handling form inputs whose values change over time along with Validations.

• Component: Angular components are a subset of directives, always associated with a template.

• Lifecycle hooks: ngOnInit()

• Data Binding: Data binding deals with how to bind your data from component to HTML DOM elements (Templates). We can easily interact with applications without worrying about how to insert our data. We can make connections in two different ways “1-way” and “2-way binding”.

• Directives: In Angular, Directives are defined as classes that can add new behavior to the elements in the template or modify existing behavior.

• Routing: The process of defining the navigation element and t associated view is called routing in Angular. Angular provides a separate module, the Router module, for setting up navigation in an Angular application.

• Services: Angular Services is a piece of reusable code with a focused purpose. A code that you will use across multiple components in your application.

• Databases: used to store and retrieve data.

• Data Sources: used to define a set of properties required to identify and access the database.

• Collections: used some collections such as array-list to store collection of data.

• Collections: used Java8 Streams to filter and fetch a collection of data.

• Custom Exception Handling: used to catch problems that arise in the code, especially in I/O blocks.

• MVC: Micro Service is an architecture that allows the developers to develop and deploy services independently.

• Micro Services (Spring Boot): Micro Service is an architecture that allows developers to develop and deploy services independently.

• RXJS Concepts: RXJS provides an implementation of the Observable type, which is needed until the type becomes part of the language. The library also provides utility functions for creating and working with observables. Behavior Subject.

• API: API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software.

**Technologies and Tools Used in the Project**

Angular: To create Quiz Application.

Angular Material: Angular Material is a User Interface (UI) component library that developers can use in their Angular projects to speed up the development of elegant and consistent user interfaces.

Visual Studio Code: Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle.

Spring Boot: To create an E-commerce project for ICNI Bank.

Postman: To the API request and response.

MySQL: To create and manage the database.

JPA: To manage the operations for the application.

Lombok: It is the tool of the java library that was used to generate code for minimizing the unused code.

Rest-API: To create the API methods and to check the response of the object entities.

Spring Tool Suite: To write and execute the code.

Tomcat: To deploy application.

**Final Setting**: Upload files to GitHub Repository

• Open your command prompt and navigate to the folder where you have created your files.

• cd <folder path>

• Initialize repository using the following command:

• git init

• Add all the files to your git repository using the following command:

• git add .

• Commit the changes using the following command:

• git commit . -m

• Push the files to the folder you initially created using the following command:

• git push -u origin master

**Unique Selling Points:**

1. This application is use to store & manage the Company products, users & Purchase Record

2. This application will keep on running until user close the program.

3. It allows the admin to add, update , delete, for the products, Users & Purchase Record.

4. It allows the admin to Update Password, Search Reports by date and Category & Search user by email.

**Github Link:**

**https://github.com/Neeti-Kulkarni/ICIN-Bank**

**Conclusion**:

Further enhancements to the application can be made which may include to deliver better Front-end.