04.06.2022

ICIN BANK CAPSTONE PROJECT

Prepared by: PRIYANKA DAS

Project Objective :

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

Background of the problem statement :

ICIN is one of the top banking firms that accepts deposits from the public for the purpose of lending loans to the public. It also invests an amount in securities.
Recently, the business analysts noticed a drop in the number of customers of the bank. They found out that online banking systems of banks like AXIS and American Express are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer who can develop an online banking web application with a rich and user-friendly interface.
You are hired as one of the Full Stack Java developers and have been asked to develop the web application. The management team has provided you the requirements and their business model so that you can easily arrange different components of the application.

Features of the Application :

1. Registration
2. Login
3. Account transactions
4. Transfers
5. Savings details
6. Profile settings
7. Requesting cheque books

Recommended Technologies :

1. Database management: MySQL
2. Back-end logic: Java programming, SpringBoot framework
3. Front-end development: Angular 2, HTML/CSS,
4. Automation and testing technologies: Selenium and JUnit
5. DevOps and production technologies: Git, GitHub, Jenkins, Docker, and AWS

User Portal :

It deals with the user activities. The user should be able to:

- Register or log in to the application to maintain a record of activities
- Deposit and withdraw money from the account
- View transactions and balance in the primary and savings account
- Transfer funds between different accounts and add recipients
- Request cheque books for different accounts

Admin Portal :

It deals with all the back-end data generation and product information. The admin user should be able to:

- Authorize the roles and guidelines for the user
- Grant access to the user regarding money transfer, deposits, and withdrawal
- Block the user account in case of any threat
- Authorize the cheque book requests

Accomplishments Of The Application :

ICIN is one of the top banking firms that accepts deposits from the public for the purpose of lending loans to the public. It also invests an amount in securities. Recently, the business analysts noticed a drop in the number of customers of the bank. They found out that online banking systems of banks like AXIS and American Express are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer who can develop an online banking web application with a rich and user-friendly interface.

Project Guidelines :

- The project will be delivered within four sprints with every sprint delivering a minimal viable product.
- It is mandatory to perform proper sprint planning with user stories to develop all the components of the project.
- The learner can use any technology from the above-mentioned technologies for different layers of the project.
- The web application should be responsive and should fetch or send data dynamically without hardcoded values.
- The learner must maintain the version of the application over GitHub and every new change should be sent to the repository.
- The learner must implement a CI/CD pipeline using Jenkins.
- The learner should also deploy and host the application on an AWS EC2 instance.
- The learner should also implement automation testing before the application enters the CI/CD pipeline.
- The learner should use Git branching to separately perform the basic automation testing of application.
- The learner should make a rich front-end of the application, which is user- friendly and easy for the user to navigate through the application.
- There will be two portals in the application, namely the admin and user portal.

Setup Guide :

- TemplateApp
- This project was generated with Angular CLI version 10.0.1.
- Development server
- Run ng serve for a dev server. Navigate to http://localhost:4200/. The app will automatically reload if you change any of the source files.
- Code scaffolding
- Run ng generate component component-name to generate a new component. You can also use ng generate directive|pipe|service|class|guard|interface|enum|module.
- Build
- Run ng build to build the project. The build artifacts will be stored in the dist/ directory. Use the --prod flag for a production build.
- Running unit tests
- Run ng test to execute the unit tests via Karma.
-

Setup Guide :

- Running end-to-end tests
- Run ng e2e to execute the end-to-end tests via Protractor.
- Further help
- To get more help on the Angular CLI use ng help or go check out the Angular CLI README.
- AngularApp
- This project was generated with Angular CLI version 10.0.1.
- Development server
- Run ng serve for a dev server. Navigate to http://localhost:4200/. The app will automatically reload if you change any of the source files.
- Code scaffolding
- Run ng generate component component-name to generate a new component. You can also use ng generate directive|pipe|service|class|guard|interface|enum|module.
-

Setup Guide :

- Build
- Run ng build to build the project. The build artifacts will be stored in the dist/ directory. Use the --prod flag for a production build.
- Running unit tests
- Run ng test to execute the unit tests via Karma.
- Running end-to-end tests
- Run ng e2e to execute the end-to-end tests via Protractor.
- Further help
- To get more help on the Angular CLI use ng help or go check out the Angular CLI README.
- © 2022 GitHub, Inc.
- Terms
- Privacy
- Security
- Status