**JAVA-FSD (PHASE-7)**

**CAPSTONE PROJECT – ICIN BANK**

Developer details:-

Ritik

Technical Trainee

TEK System Global Services

Link to this project: <https://github.com/Rkcr7/ICIN-BANK.git>

Instructed By: Mr. Radhakrishnan

**TITLE OF PROJECT:       ICIN Bank**

**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 with 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 chequebooks

**Recommended technologies:**

                    1. Database management: MySQL

                    2. Back-end logic: Java programming, Spring Boot framework

                    3. Front-end development: Angular, HTML/CSS,

                    4. Testing technologies: Junit

                    5. DevOps and production technologies: Git, GitHub, Jenkins, Docker, and AWS

**Project development 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 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. More information on this is on the next page.

**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 transfers, deposits, and withdrawal

                  ● Block the user account in case of any threat

                  ● Authorize the chequebook requests

**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

                ● Cheque book requests for different accounts

**Technology Used**

* Eclipse - Project Development
* JIRA –Sprint Planning
* Git Repository
* Technology : Java ,SpringBoot, Angular 2,REST , Security-spring security,MySQL
* Database,ORM,Hibernate,logger,HTML,CSS,JavaScript
* Framework: Spring MVC,Spring Boot
* Server: tomcat
* Repository: Git
* GitHub Url: https://github.com/Rkcr7/ICIN-BANK.git

|  |
| --- |
| **1.Project Description from LMS:**  Based on the Requirements from the LMS, I developed the ICIN Bank website application prototype, Where website must have work for User Login as well as for Admin Login .  **User can access the below functionality:**  Accounts   * Primary   + View Transaction list   + Search Transaction * Savings   + View Transaction list   + Search Transaction   Transfer   * Between Accounts * Person-To-Person * Add/Edit Recipients   Transaction   * Deposit * Withdrawal   Menu   * Profile Settings * Schedule Appointment for requesting Chequebook   Registration  **Admin have the below functionality access :**   * View user * View user transaction list * Enable/Disable user account * View Appointment * Confirm appointment   .  **Sprints planning :**   1. MySql setup and create all the required tables for the ICIN Bank. 2. Create SpringBoot Application for buliding microservices running in the backend ie : APIs. 3. Setting up Frontend with HTML,CSS and Angular 4. Creating Applicationg for both Admin and User interface for operating ICIN bank. 5. Creating images of BackendApi, Admin UI, User UI and upload in GitHub.   The flow of the Application   1. Data flow diagram      1. Architecture Design     Modules in the Project:  1. Admin Login  2. User Login  3. Change password  4. Logout  5. Register User  6. Account summary  7. Transaction summary  8. Account maintenance  9. Beneficiary maintenance  10. Cheque Book requests & approvals  11. User access maintenance  **Technologies used for these features are SringBoot,REST,Security,Spring MVC,Hibernate and MySQL.**   * Java Collection Framework , foreach loop ,jsp tags ,POST and GET method,html ,web services has been used as core concept,REST,Angular,JS,CSS,HTML,Hibernate,Springboot,Security.   URL for Users: <http://localhost:8080/index>  URL For User SignUp: <http://localhost:8080/signup>  URL for for User Signin: <http://localhost:8080/signin>  URL for admin: <http://localhost:4200/login> ,login as admin on this url also-<http://localhost:8080/index>  **Database:**    Below are the Test Cases:   1. Signup     2.If user`s email is already exist    **User Signin Page**-  **Url**- <http://localhost:8080/index>      **After Signin user homepage-**    **Primary Account Details-**     1. **Saving Account Detail-**      1. **Deposit Functionality-**     **Withdraw Service for User-**    **Transfrer Between Accounts-**    **Add/Edit Recipient-**    **Transfer to someone else Account-**    **User can request chequebook by taking appointment with bank-**      **User Profile functionality where user can change his/her profile information-**    **Admin Portal Module-**   1. **Admin login url->**<http://localhost:4200/login>   **\*Note:**after login from Admin Portal url admin needs to login in <http://localhost:8080/index> url also.  **2. Whenever you signup from** <http://localhost:8080/index> url , if you want you can give user role as admin or user from database. Currently admin role is assigned to **username**:admin  **Password:**admin  3.Once admin has logged in admin screen will open where admin can see the user list with user’s account details and admin can make user enable or disable and can confirm appoint for chequebook also.  **Admin portal screen-**    **Admin can see User Account Detail from User Account Tab-**    **Admin can check Primary and Savings Transaction details by clicking on account balance of Primary Account and Saving Account-** |
| **Admin can See the Chequebook Request in Appoint Tab for Chequebooks and Confirm fromAction column once he click on confirm Confirmed column will become true from false-**    **My SQL DB-**  **username:root**  **password:root** |

**REFERENCES**

1. <https://angular.io/>
2. <https://www.google.com/>
3. <https://spring.io/projects/spring-boot>
4. <https://docs.oracle.com/en/java/>