

// Algorithm for mAadhar Application

1. Begin

2. Design the database schema to store user information, Aadhar application details

3. Backend

Implement RESTful APIs for registration, login, applying for a new Aadhar card, updating Aadhar details, applying for a duplicate Aadhar card, and applying to close Aadhaar card (due to death).

Use Spring Boot to simplify API development.

Utilize JPA and Hibernate for object-relational mapping to interact with the MySQL database.

4. Frontend

Develop a user-friendly web application where users can register, login, and perform other Aadhar-related operations.

Use Angular as the frontend framework to build dynamic and responsive web pages.

Utilize Bootstrap for easy styling and layout.

Use HTML/CSS for designing the user interface.

5. Admin Portal

- **Login through admin credentials**
- **Approve new Aadhaar Card request**
- **Verify request for duplicate Aadhaar**

- **Display all issued Aadhaar Card**
- **Delete Aadhaar card details for dead citizen**

6. User Portal

- **Sign in to apply for a new Aadhar Card**
- **Login to see the Aadhar number assigned by the admin**
- **Update address, phone number, and date of birth of Aadhaar**
- **Request duplicate Aadhaar Card**

7. Automation and Testing

Implement test automation using Selenium and TestNG for functional and integration testing.

Write test cases to validate different functionalities of the application

8. DevOps

Use Git for version control to manage the source code.

Host the code repository on GitHub for collaboration and easy access.

Set up a Jenkins pipeline for continuous integration and deployment.

Use Docker to containerize the application for easier deployment and scalability

9. End