The home page of the Aadhar Card application serves as the main entry point for users and provides an overview of the application's key features and functionalities. Its purpose is to welcome users, guide them through the application, and prompt them to take specific actions. Here's a description of what the home page might contain:

Application Logo and Name: The top section of the home page should display the application's logo and name, providing users with quick identification.

Welcome Message: Below the logo, include a warm and welcoming message that greets users as they visit the application. For example, "Welcome to Varniraj Aadhar Card Application."

Brief Overview: Provide a concise overview of the application's purpose and benefits. This section should highlight the convenience and ease of applying for an Aadhar Card using the platform.

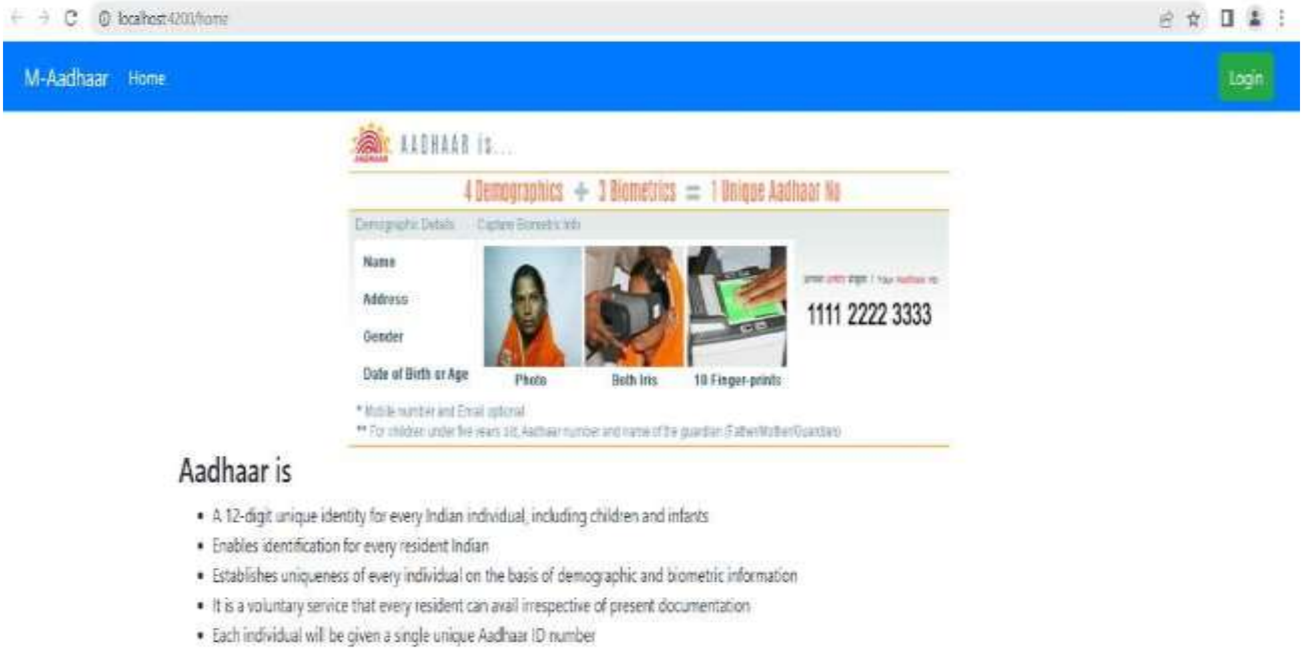
Key Features: List the key features of the application in bullet points or brief paragraphs. Mention functionalities like user registration, login, application for new Aadhar Cards, updating Aadhar details, applying for a duplicate Aadhar Card, and closing Aadhaar cards due to unfortunate events.

CTA Buttons: Add clear Call-to-Action (CTA) buttons to prompt users to take specific actions. Examples of CTA buttons could be "Apply Now," "Login," "Register," etc. Each button should lead users to the respective pages for performing the action.

Images or Illustrations:

Incorporate relevant images or illustrations that represent the Aadhar Card application process or the benefits of having an Aadhar Card. This can make the home page visually appealing and engaging.

User Testimonials (Optional): If available, consider adding a section for user testimonials or reviews. Positive feedback from satisfied users can instill trust and confidence in the application's credibility.



Aadhaar is

- A 12-digit unique identity for every Indian individual, including children and infants
- Enables identification for every resident Indian
- Establishes uniqueness of every individual on the basis of demographic and biometric information
- It is a voluntary service that every resident can avail irrespective of present documentation
- Each individual will be given a single unique Aadhaar ID number

Sign Up Section Title: The section should start with a clear and prominent title like "Sign Up" to indicate that this is the registration area.

User and Admin Tabs:Display two tabs, one for "User" and the other for "Admin," allowing users to choose their role during the registration process. These tabs help differentiate between regular users and administrative users.

Registration Form:Below the tabs, provide a registration form with fields for users to enter their email and password.

Email Field: A text input box where users can enter their email address. This field should be labeled appropriately, like "Email" or "Email Address."

Password Field: A password input box where users can enter their chosen password. Include a password visibility toggle to allow users to see their entered password.

Confirm Password Field: An additional password input box where users need to re-enter their password to confirm accuracy.

Submit Button: Add a "Sign Up" or "Register" button at the bottom of the form to initiate the registration process.

Validation and Error Handling:Implement client-side validation for email and password fields. Display error messages if the entered information is not valid or required fields are empty.

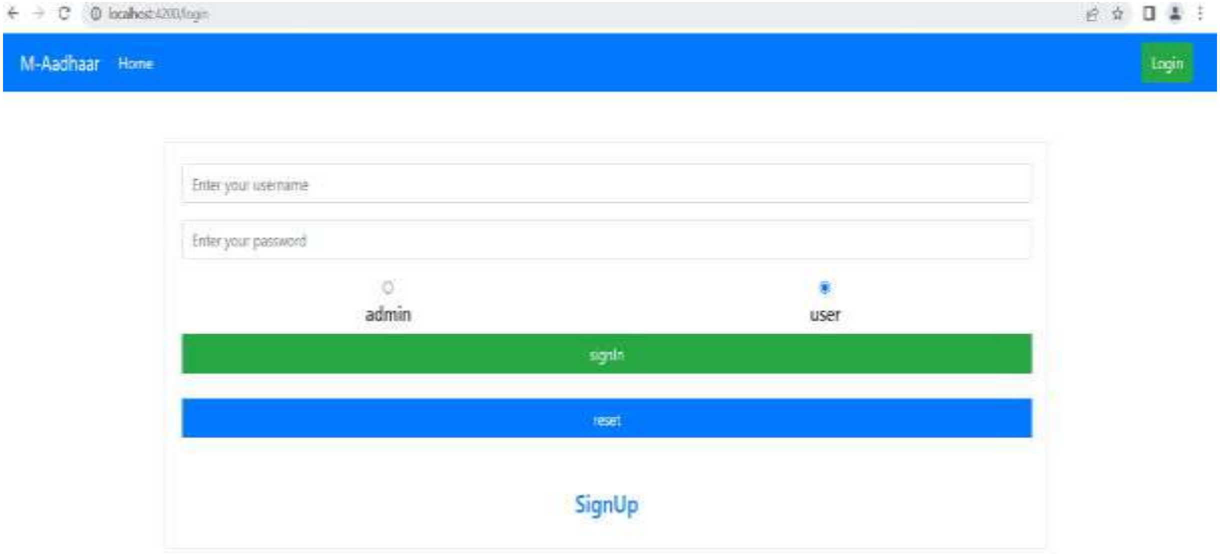
Server-side validation is also crucial to ensure data integrity and security.

Role Selection:Upon selecting the "User" tab, the user should be able to register as a regular user with standard permissions.

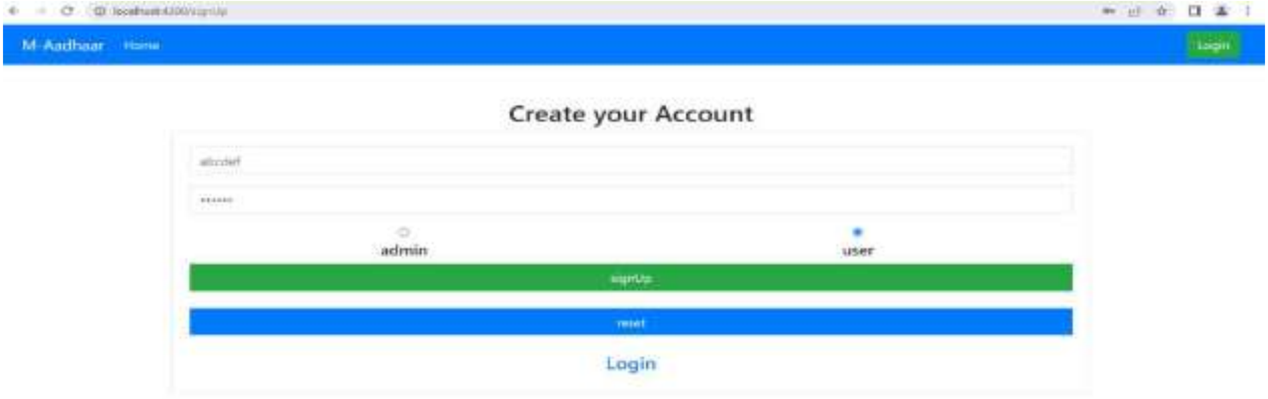
If the "Admin" tab is chosen, additional admin-specific fields or steps can be displayed to gather more information about the admin user (e.g., admin ID, organization, etc.).

Terms and Conditions Checkbox:Include a checkbox for users to agree to the application's Terms and Conditions and Privacy Policy before proceeding with the registration.

Password Strength Indicator (Optional):If you want to enhance security, consider adding a password strength indicator to guide users in creating strong and secure passwords.



The "Create Account in Aadhar" feature allows Indian citizens to register and apply for an Aadhar Card through an automated and streamlined process. Users provide their personal details, such as name, date of birth, address, and mobile number, during the account creation process. They also select a unique username and password for secure access to their account. Upon successful registration, users receive a unique Aadhar number that serves as a valid identification proof issued by the Government of India. The application ensures data privacy and security by adhering to standard encryption practices. With their Aadhar account, users can conveniently access services, update their Aadhar details, and request a duplicate Aadhar Card if needed.

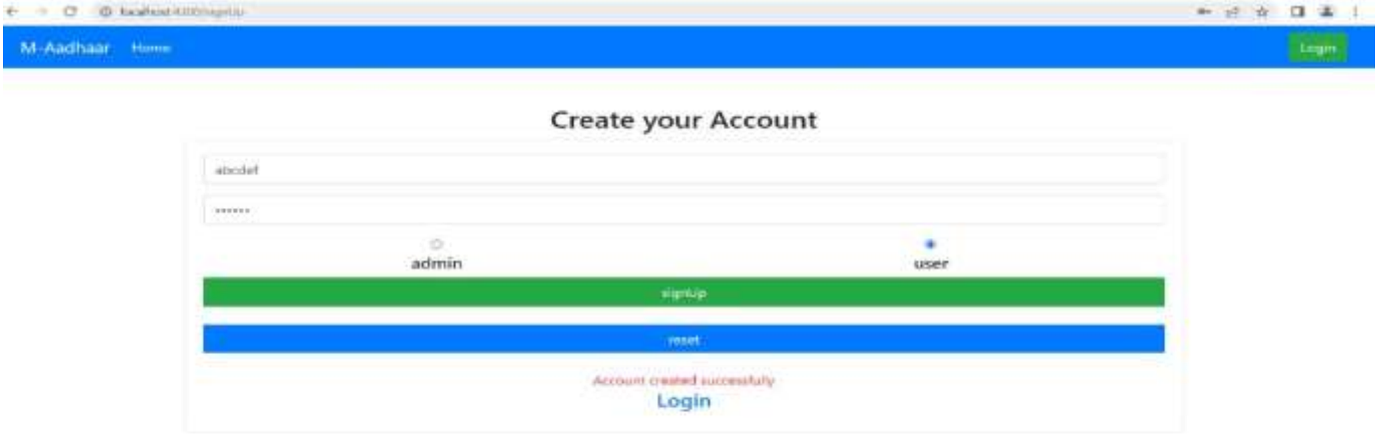


Create your Account

abcdef

******

○ admin ● user



Create your Account

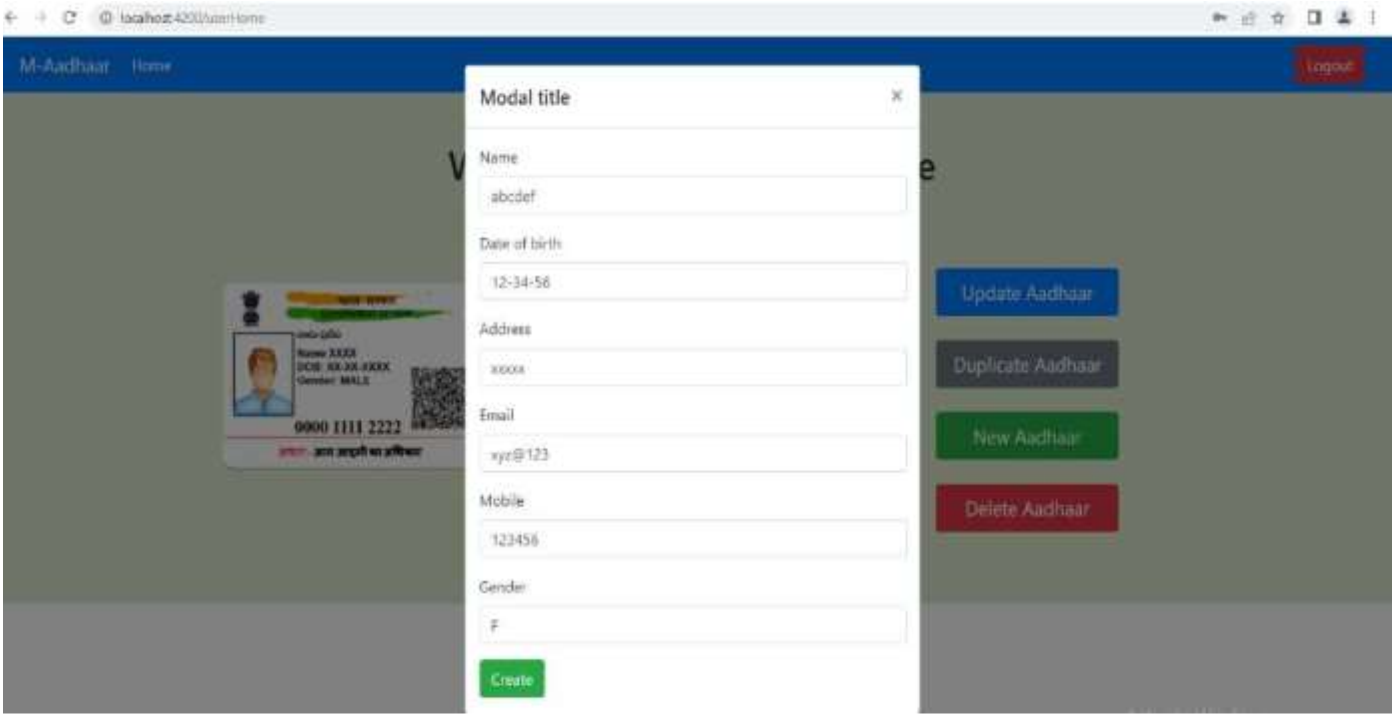
abcdef

******

○ admin ● user

Account created successfuly

After a user successfully registers and creates an account in the Aadhar application, it is essential to provide a reassuring and positive feedback message to confirm their registration. The "Account Created Successfully" message is displayed to acknowledge the user's successful completion of the account creation process. This message should be clear, concise, and prominently displayed on the user interface. Here's a description of the message: Alert Message Title: "Account Created Successfully"



Modal title

Name: abcdef

Date of birth: 12-34-56

Address: xxxxx

Email: xyz@123

Mobile: 123456

Gender: F

Upon successful login or account creation, users are greeted with the "Welcome to M Aadhar" page. This page serves as the central hub for all Aadhar-related services, providing easy access to various functionalities. Here's a description of the services available on the "Welcome to M Aadhar" page:

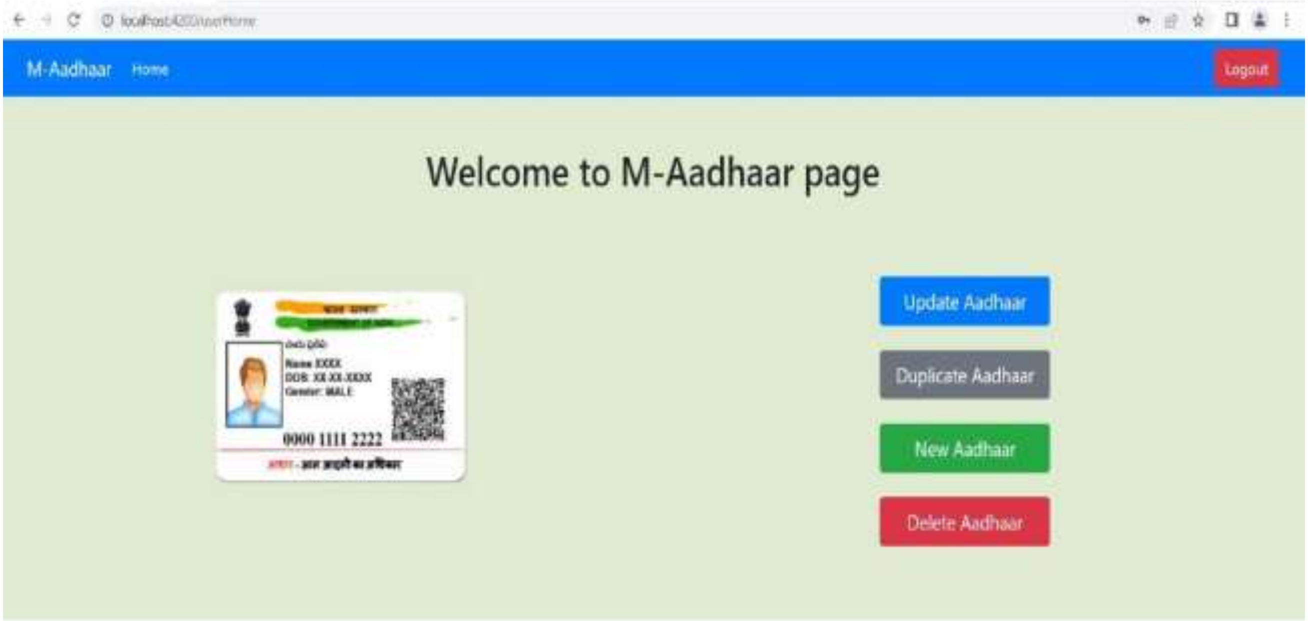
Update Aadhar Details: Description: "Keep your Aadhar information up-to-date by using this service. Update your personal details such as address, mobile number, and biometric data. Ensure the accuracy of your Aadhar details to maintain the validity and effectiveness of your Aadhar card as an identification proof."

Duplicate Aadhar Card: Description: "Lost or misplaced your Aadhar Card? No worries! Request a duplicate Aadhar card through this service. In case your original Aadhar card is damaged or stolen, obtain a new copy hassle-free. We'll issue a duplicate Aadhar card with the same unique Aadhar number."

New Aadhar Card Application: Description: "If you don't have an Aadhar card yet, this is the place to apply for one. Get started with the application process by providing your essential details. Once your application is processed, you'll receive a unique Aadhar number issued by the Government of India."

Delete Aadhar Card (Due to Death): Description: "In unfortunate circumstances of an individual's passing, use this service to apply for Aadhar card closure. By providing necessary documentation, we will deactivate the Aadhar card and mark it as closed. This process ensures data security and prevents misuse."

Note: Here description provided is sample we can add it but in our project it was not added.



On the Aadhar Registration Page, Indian citizens can begin the process of obtaining their unique Aadhar Card. The page gathers essential personal information required for the registration. Here's a description of the fields and the purpose of the "Create" button:

Name: Enter your full name as per official documents. This ensures accuracy in the Aadhar Card.

Date of Birth: Provide your date of birth in the format DD/MM/YYYY. It is crucial for age verification.

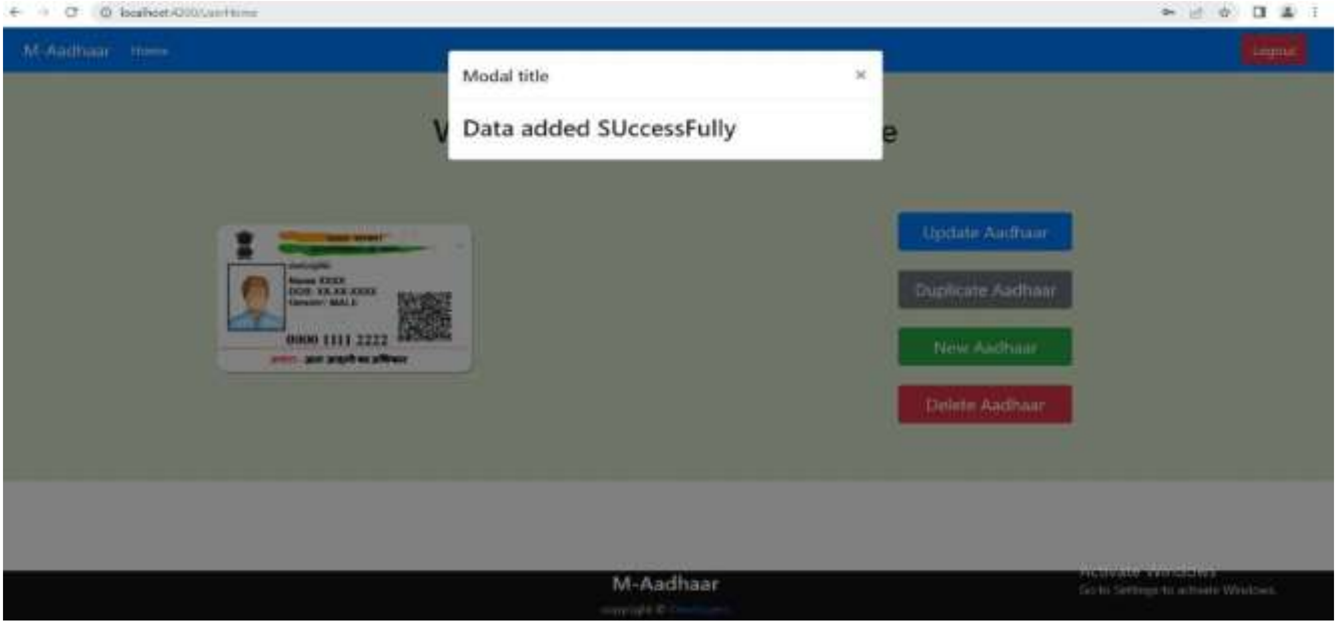
Address: Enter your current residential address. Make sure the address is accurate as Aadhar serves as proof of residence.

Email: Provide a valid email address. This will be used for communication and updates related to your Aadhar application.

Mobile: Enter your active mobile number. An OTP will be sent to this number for verification during the registration process.

Gender: Choose your gender from the available options (Male, Female, or Other).

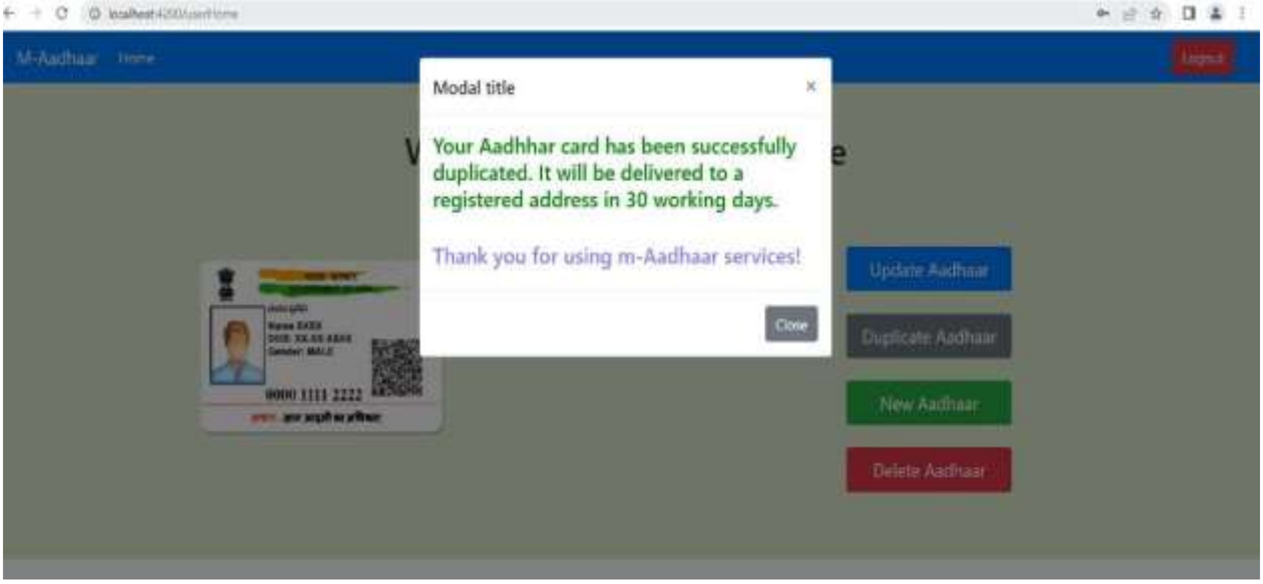
"Create" Button: Description: Click the "Create" button to initiate the Aadhar registration process. Once you click this button, your provided information will be verified and stored securely in the Aadhar database. You will receive an OTP on the provided mobile number for verification purposes. Follow the on-screen instructions to complete the registration process successfully.



Modal title

Data added SUccessFully

After successfully completing the Aadhar registration process, users should receive a reassuring message confirming that their data has been added to the system. This message serves as a positive acknowledgment of their successful registration. Here's the description of the "Data Added Successfully" message:



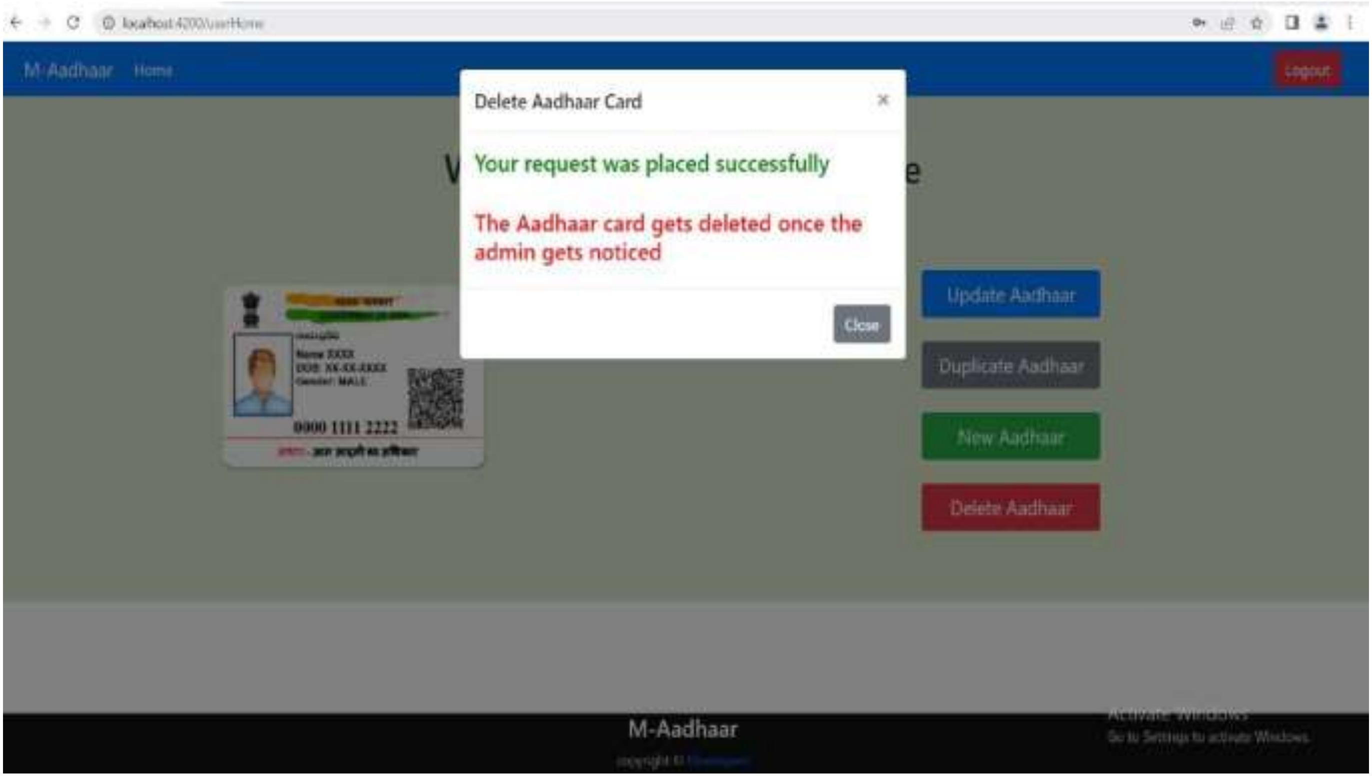
Modal title

Your Aadhhar card has been successfully duplicated. It will be delivered to a registered address in 30 working days.

Thank you for using m-Aadhaar services!

After successfully submitting the request for a duplicate Aadhar Card, users should receive a confirmation message acknowledging the successful duplication process. This message provides important details regarding the delivery of the duplicate Aadhar Card. Here's the description of the "Duplicate Aadhar Card Request Successful" message:

Description: "Congratulations! Your request for a duplicate Aadhar Card has been successfully processed. The duplicate Aadhar Card will be delivered to your registered address within 30 working days. Please keep an eye on your mailbox for its arrival. Should you have any questions or concerns, our support team is available to assist you. Thank you for using M Aadhar services, your trusted Aadhar Card solution."



Delete Aadhaar Card

Your request was placed successfully

The Aadhaar card gets deleted once the admin gets noticed

Close

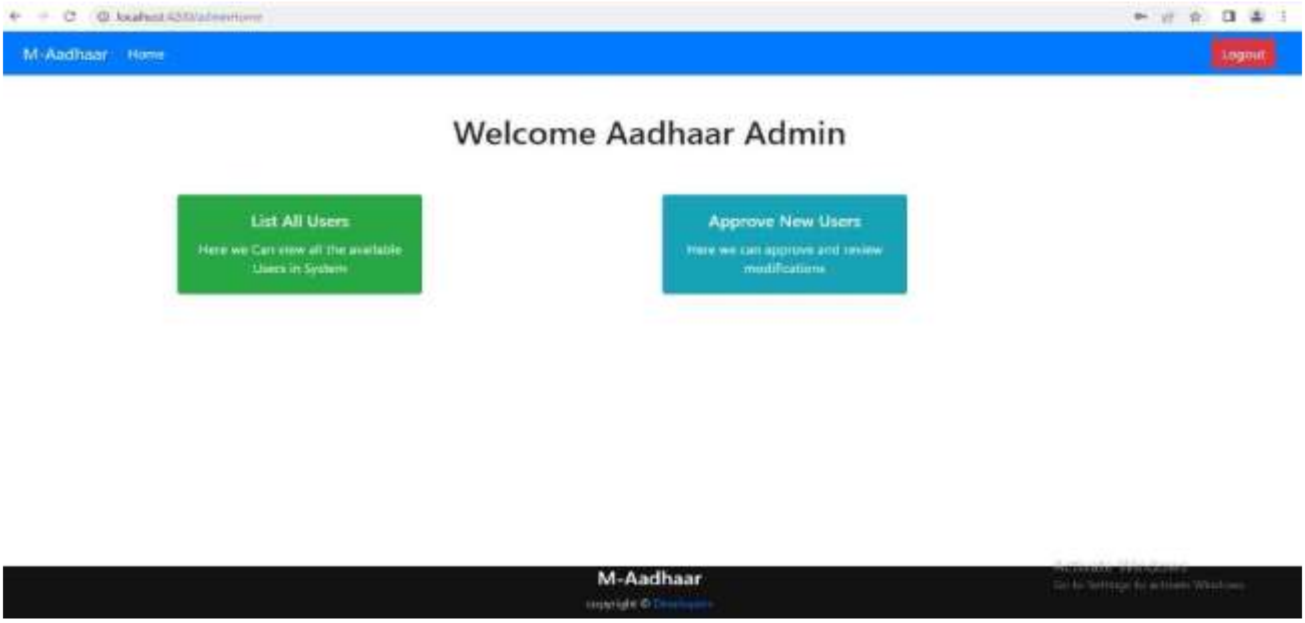
Deleting an Aadhar card is not a standard feature, as Aadhar cards are unique identification numbers issued by the Government of India, and once generated, they cannot be deleted. The Aadhar card database is maintained securely by the government authorities to ensure data integrity and prevent misuse.

However, there are specific scenarios where an individual's Aadhar card may need to be deactivated or marked as "Inactive." Such situations can include:

Deceased Individual: In the unfortunate event of an individual's death, their Aadhar card may be deactivated to prevent any misuse of their identity.

Duplicate Aadhar Card: If it is discovered that an individual has been issued multiple Aadhar cards due to errors or data duplication, one of the cards may be deactivated.

In such cases, the deactivation or marking of an Aadhar card as "Inactive" is done by the relevant government authorities, and the individual or their family members need to approach the appropriate Aadhar enrollment center or authorities to initiate the process.



Welcome Aadhaar Admin

List All Users

Here we Can view all the available Users in System

Approve New Users

Here we can approve and review modifications

M-Aadhaar

copyright © Developers

The "Welcome Aadhar Admin" page is a dashboard designed specifically for Aadhar administrators to manage user registrations and approvals efficiently. This page provides access to a list of users who have registered for Aadhar and allows the admin to approve new user registrations. Here's a description of the page elements:

Title: "Welcome Aadhar Admin"

The page should start with a prominent title welcoming the Aadhar admin to the dashboard.

List of Users: Display a table or list of registered users who have applied for Aadhar Card registration. This list includes essential user details such as name, date of birth, address, email, and mobile number.

User Status: Next to each user in the list, indicate their current status, such as "Pending Approval," "Approved," or "Rejected."

"Approve" Button: Add an "Approve" button beside each user entry in the list to allow the admin to approve new user registrations. When clicked, this button confirms the user's details and generates a unique Aadhar number.

"Reject" Button (Optional):Optionally, include a "Reject" button beside each user entry to reject the application if the provided information is incorrect or incomplete.

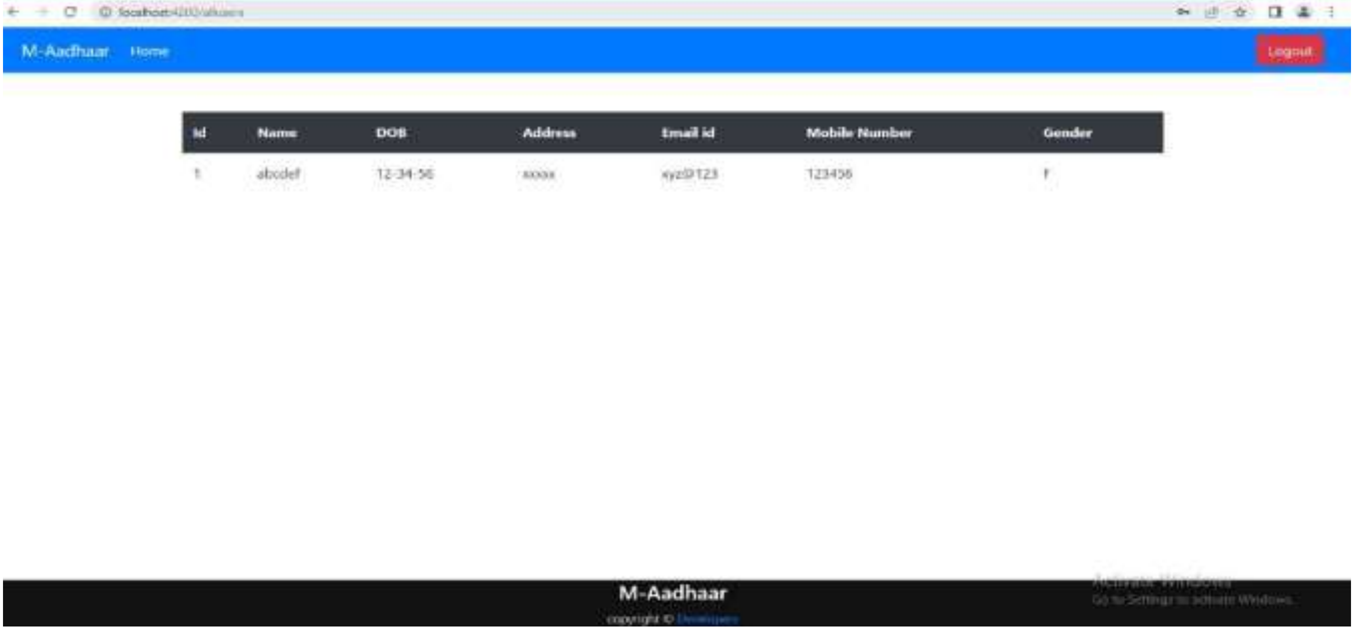
Search and Filter:

Provide search and filter options to allow the admin to quickly find specific users based on their details or status.

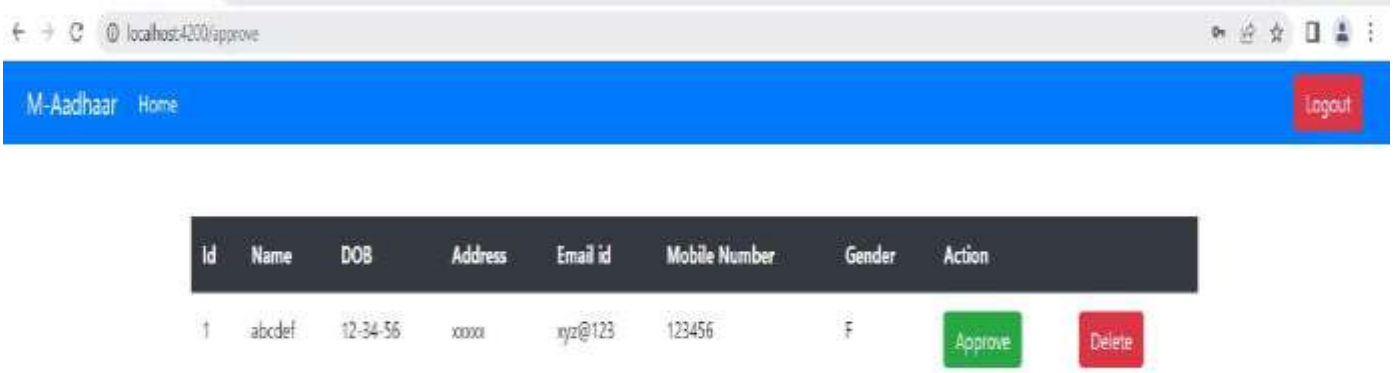
Pagination:If there are many registered users, implement pagination to display a limited number of users per page for easy navigation.

"Logout" Button:Place a "Logout" button in the top right corner to allow the admin to log out from the dashboard.

Help and Support:Offer a help section or support contact information for the admin to seek assistance or report any issues.



Id	Name	DOB	Address	Email id	Mobile Number	Gender
1	abcdef	12-34-56	xxxxx	xyz@123	123456	F



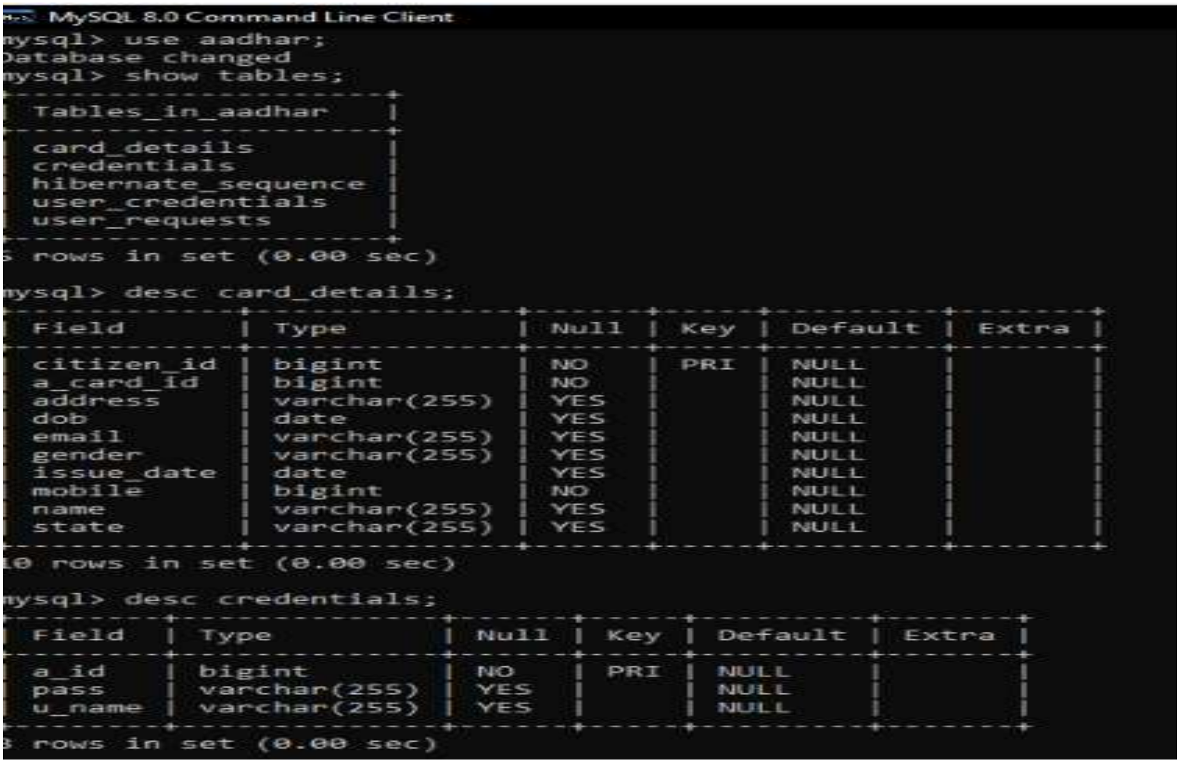
Id	Name	DOB	Address	Email id	Mobile Number	Gender	Action	
1	abcdef	12-34-56	xxxxx	xyz@123	123456	F	Approve	Delete

Table: users Purpose: This table stores user information, including details like user ID, username, password (hashed and salted), name, date of birth, address, email, mobile number, and gender.

Table: aadhar_cards Purpose: This table stores the details of each Aadhar card issued to users. It includes the Aadhar number, user ID (foreign key referencing the users table), issue date, and status (active, inactive, or closed).

Table: admin_users Purpose: This table stores information about Aadhar administrators. It includes details such as admin ID, username, password (hashed and salted), name, and role.

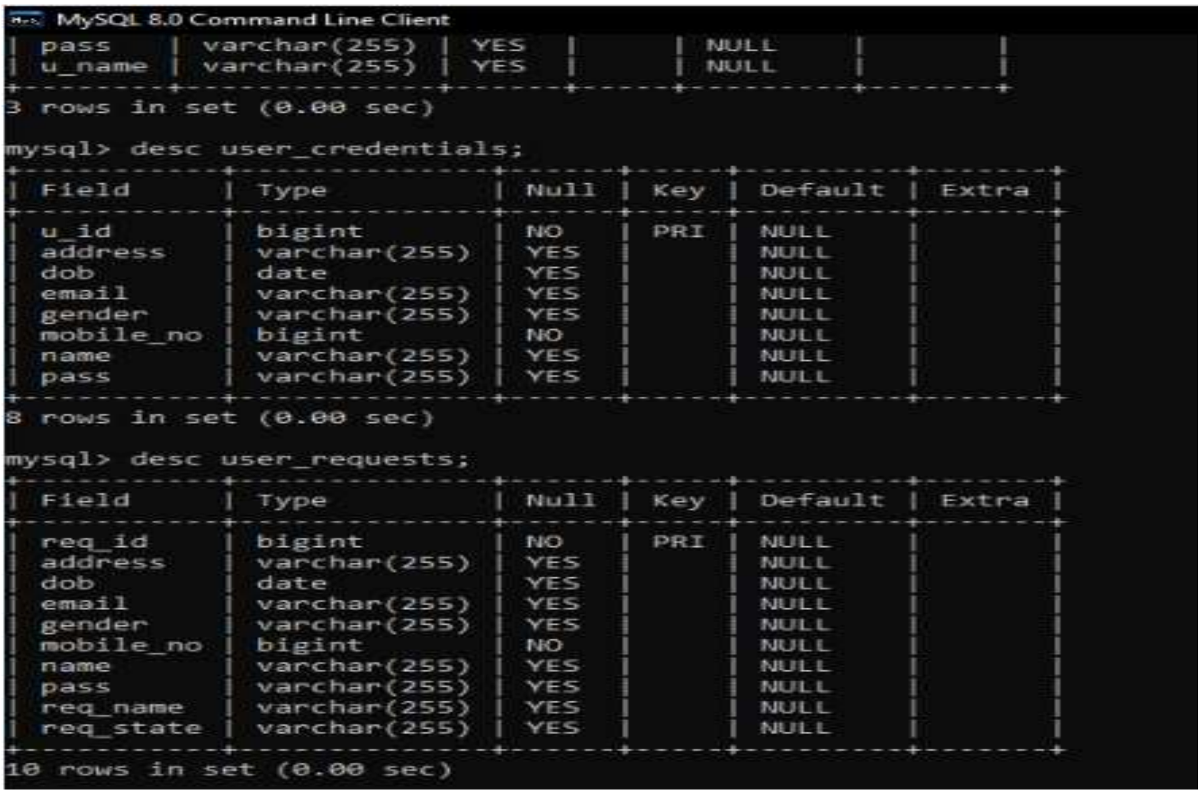
Table: aadhar_applications Purpose: This table stores the applications submitted by users for new Aadhar cards or updates. It includes application ID, user ID (foreign key referencing the users table), application type (new, update, duplicate, etc.), application date, status (pending, approved, rejected), and any additional details required for the application.



```
ysql> use aadhar;
atabase changed
ysql> show tables;
+---------------------+
| Tables_in_aadhar    |
+---------------------+
| card_details        |
| credentials         |
| hibernate_sequence  |
| user_credentials    |
| user_requests       |
+---------------------+
5 rows in set (0.00 sec)

ysql> desc card_details;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| citizen_id  | bigint       | NO   | PRI | NULL    |       |
| a_card_id   | bigint       | NO   |     | NULL    |       |
| address     | varchar(255) | YES  |     | NULL    |       |
| dob         | date         | YES  |     | NULL    |       |
| email       | varchar(255) | YES  |     | NULL    |       |
| gender      | varchar(255) | YES  |     | NULL    |       |
| issue_date  | date         | YES  |     | NULL    |       |
| mobile      | bigint       | NO   |     | NULL    |       |
| name        | varchar(255) | YES  |     | NULL    |       |
| state       | varchar(255) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
10 rows in set (0.00 sec)

ysql> desc credentials;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| a_id   | bigint       | NO   | PRI | NULL    |       |
| pass   | varchar(255) | YES  |     | NULL    |       |
| u_name | varchar(255) | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```



```
| pass   | varchar(255) | YES  |     | NULL    |       |
| u_name | varchar(255) | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> desc user_credentials;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| u_id      | bigint       | NO   | PRI | NULL    |       |
| address   | varchar(255) | YES  |     | NULL    |       |
| dob       | date         | YES  |     | NULL    |       |
| email     | varchar(255) | YES  |     | NULL    |       |
| gender    | varchar(255) | YES  |     | NULL    |       |
| mobile_no | bigint       | NO   |     | NULL    |       |
| name      | varchar(255) | YES  |     | NULL    |       |
| pass      | varchar(255) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql> desc user_requests;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| req_id    | bigint       | NO   | PRI | NULL    |       |
| address   | varchar(255) | YES  |     | NULL    |       |
| dob       | date         | YES  |     | NULL    |       |
| email     | varchar(255) | YES  |     | NULL    |       |
| gender    | varchar(255) | YES  |     | NULL    |       |
| mobile_no | bigint       | NO   |     | NULL    |       |
| name      | varchar(255) | YES  |     | NULL    |       |
| pass      | varchar(255) | YES  |     | NULL    |       |
| req_name  | varchar(255) | YES  |     | NULL    |       |
| req_state | varchar(255) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

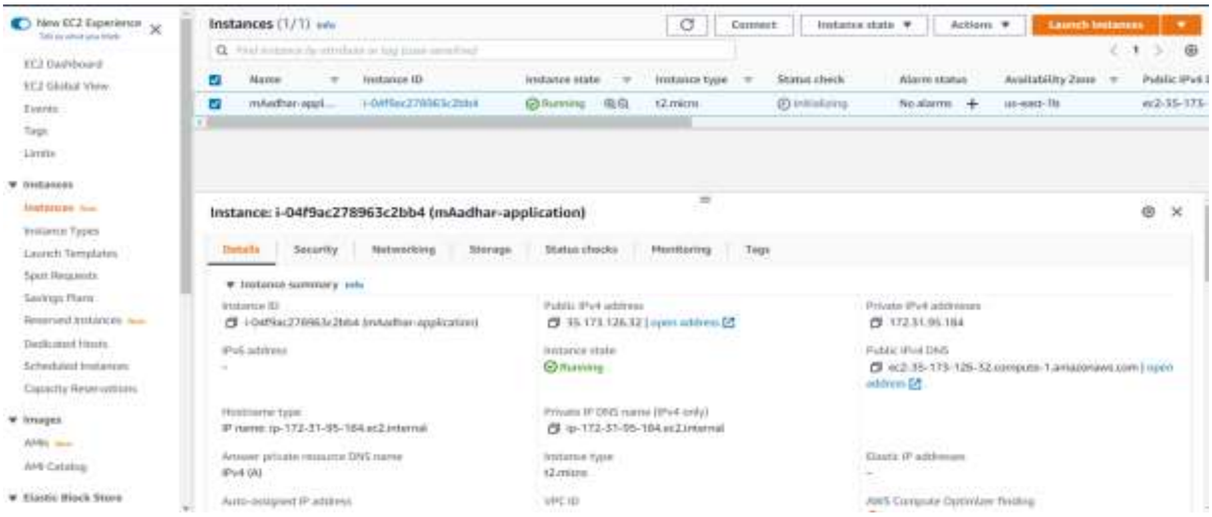
Application Availability: The M Aadhar application will be accessible to users via the public IP address or DNS name associated with the EC2 instance. Users can access the application using a web browser or any supported client.

Application Functionality: The M Aadhar application will be fully functional, allowing users to register for new Aadhar Cards, update their details, apply for duplicate Aadhar Cards, and manage their account.

Scalability: The application can easily scale up or down based on the instance type and configuration. If there is an increase in user traffic, you can add more EC2 instances using Auto Scaling to handle the load.

Performance: The application's performance will depend on factors such as the EC2 instance type, available resources (CPU, memory), and the efficiency of the application code. Properly configuring the instance and optimizing the application code will help achieve optimal performance.

This optional implementation for running server on a linux machine.



Instances (1/1)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	mAadhar-appl...	i-04f9ac278963c2bb4	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-35-173-

Instance: i-04f9ac278963c2bb4 (mAadhar-application)

Instance summary

Instance ID
i-04f9ac278963c2bb4 (mAadhar-application)

Public IPv4 address
35.173.126.32 | open address

Private IPv4 addresses
172.31.95.184

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-35-173-126-32.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-95-184.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-95-184.ec2.internal

Answer private resource DNS name
IPv4 (A)

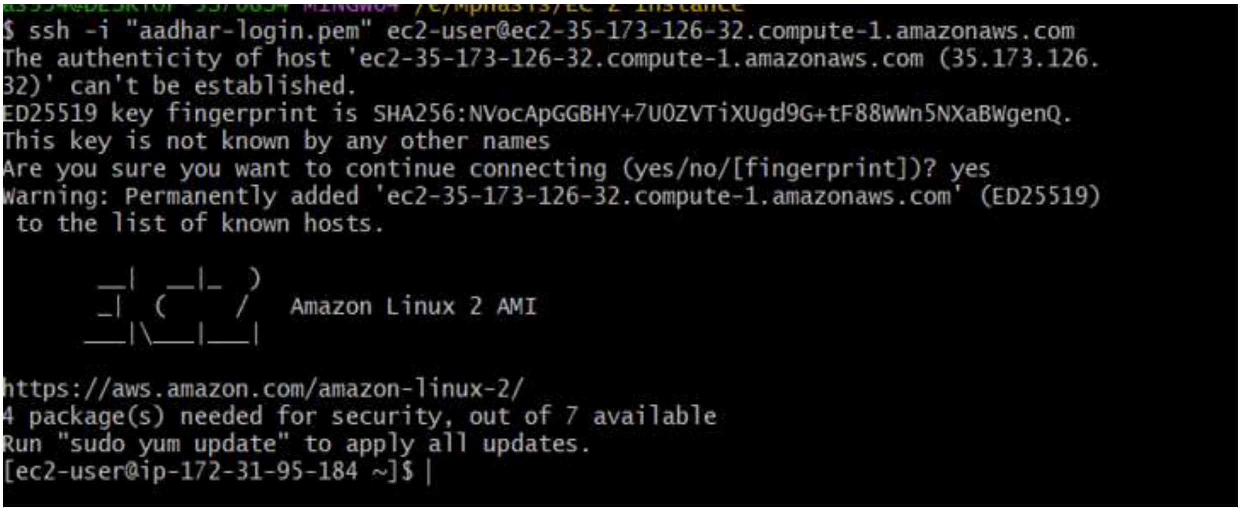
Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address

VPC ID

AWS Compute Optimizer finding

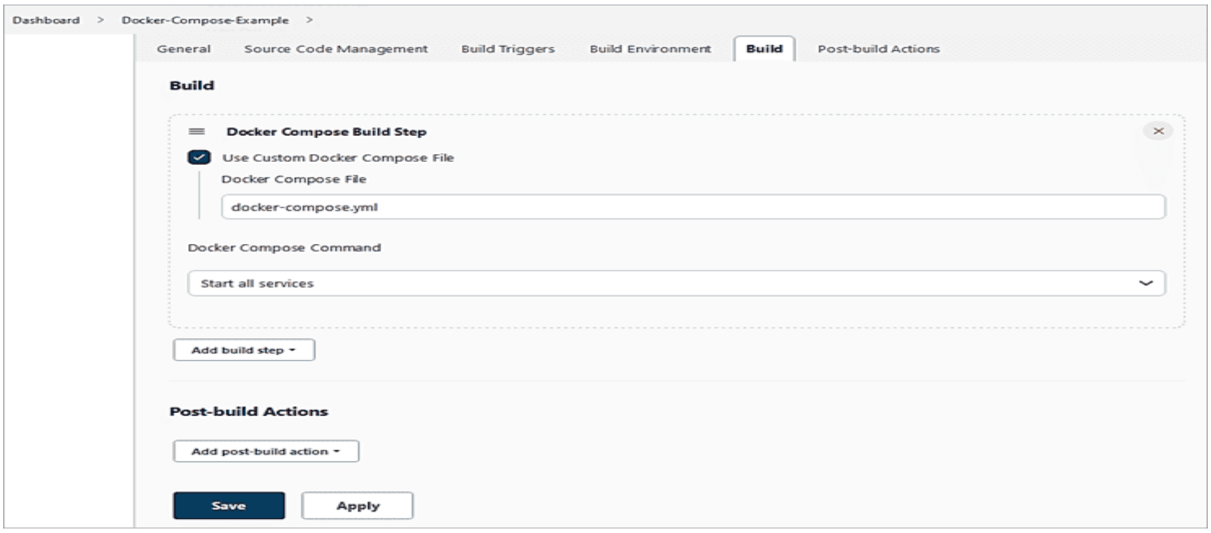


```
$ ssh -i "aadhar-login.pem" ec2-user@ec2-35-173-126-32.compute-1.amazonaws.com
The authenticity of host 'ec2-35-173-126-32.compute-1.amazonaws.com (35.173.126.
32)' can't be established.
ED25519 key fingerprint is SHA256:NVocApGGBHY+7U0ZVTiXUgd9G+tF88WWn5NXaBWgenQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-173-126-32.compute-1.amazonaws.com' (ED25519)
 to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-184 ~]$ |
```

Jenkins Job Using Docker Compose Plugin Let’s look at adding a Build step in Jenkins to call the docker-compose.yml file and start all the services defined in the file as part of the Docker Compose Command. The dockercompose.yml file is present in the root of the GitHub repository. The rest of the freestyle job configuration for SCM definition remains the same as in the previous section.


Dashboard > Docker-Compose-Example >

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Build

Docker Compose Build Step

☑ Use Custom Docker Compose File

Docker Compose File: docker-compose.yml

Docker Compose Command: Start all services

Click on Save and trigger a build. Once done, look at the console output and the services running in the docker host.

In this section, we will look at how using Jenkins as a Docker Stack can be used to deploy multiple services that are primarily containers across different machines. The services run as part of the stack can also be configured across multiple replicas. Docker stack makes use of a YAML file to deploy multiple services. In this example, I am using the below docker-compose.yml file and a 3 node cluster.

In the YAML file, I am using a built image since the build command is not supported by docker stack deploy. In Jenkins, use a Execute shell Build step and add the below commands:

docker login -u “kart1999” -p “

“ docker build -t kart1999/maadhar_tomcat:V1 .

docker push kart1999/maadhar_tomcat:V1

sleep 5

docker stack deploy -c

docker-compose.yml maadhar docker service ls

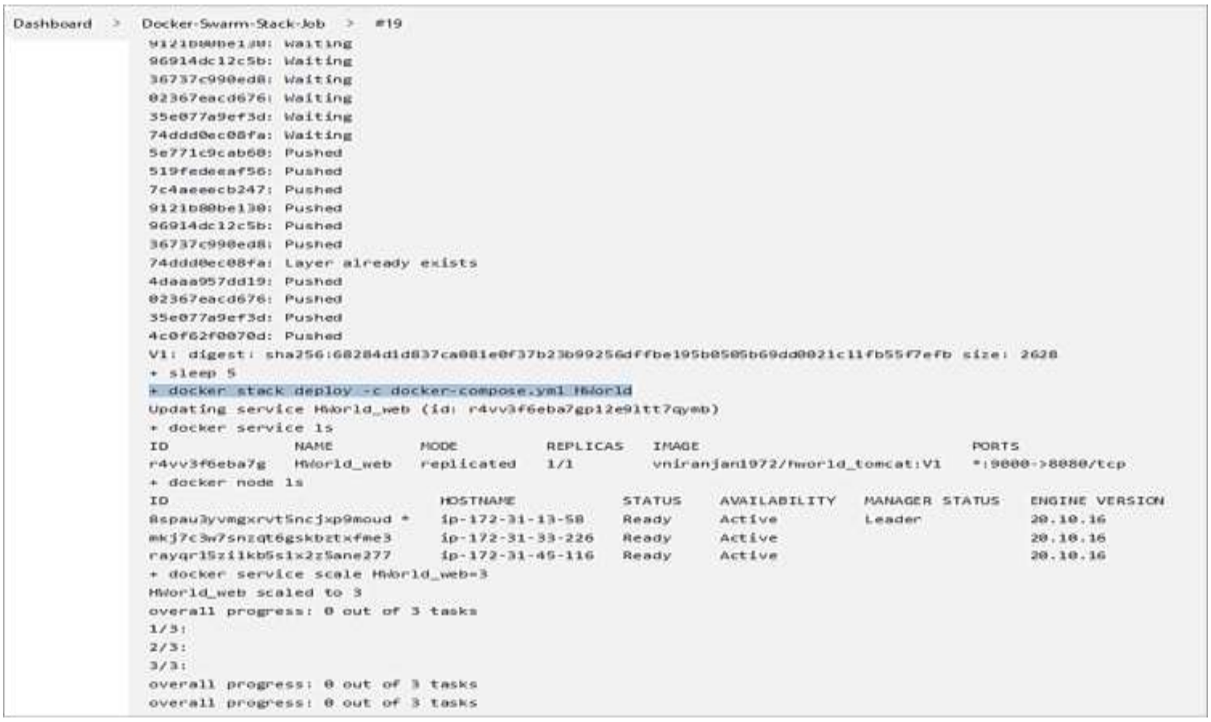
docker node ls

docker service scale maadhar_web=3

docker service ps

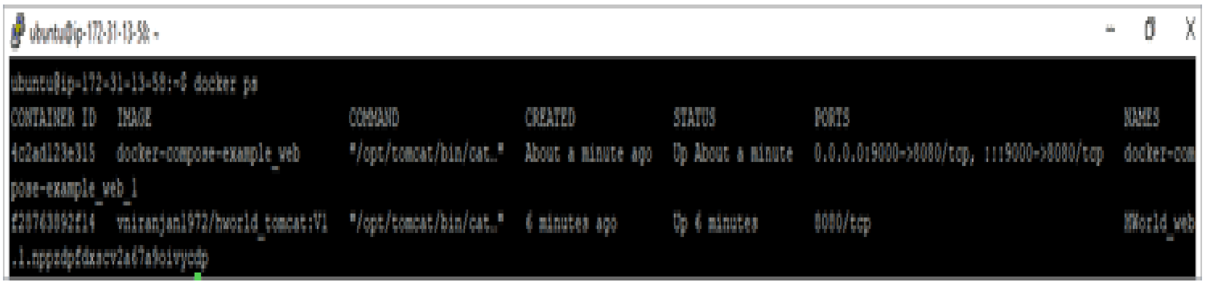
maadhar_web

Save the Job and trigger a build. The job is run on the manager node.



```
9121b80be130: Waiting
96914dc12c5b: Waiting
36737c990ed8: Waiting
02367eacd676: Waiting
35e077a9ef3d: Waiting
74ddd0ec08fa: Waiting
5e771c9cab60: Pushed
519fedeeaf56: Pushed
7c4aeeecb247: Pushed
9121b80be130: Pushed
96914dc12c5b: Pushed
36737c990ed8: Pushed
74ddd0ec08fa: Layer already exists
4daaa957dd19: Pushed
02367eacd676: Pushed
35e077a9ef3d: Pushed
4c0f62f0070d: Pushed
V1: digest: sha256:68284d1d837ca081e0f37b23b99256dffbe195b0505b69dd0021c11fb55f7efb size: 2628
+ sleep 5
+ docker stack deploy -c docker-compose.yml HWorld
Updating service HWorld_web (id: r4vv3f6eba7gp12e91tt7qymb)
+ docker service ls
ID             NAME         MODE         REPLICAS   IMAGE                              PORTS
r4vv3f6eba7g   HWorld_web   replicated   1/1        vniranjan1972/hworld_tomcat:V1     *:9000->8080/tcp
+ docker node ls
ID                            HOSTNAME           STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
8spau3yvmgxrvt5ncjxp9moud *   ip-172-31-13-58    Ready     Active         Leader           20.10.16
mkj7c3w7snzqt6gskbztxfme3     ip-172-31-33-226   Ready     Active                          20.10.16
rayqr15zi1kb5s1x2z5ane277     ip-172-31-45-116   Ready     Active                          20.10.16
+ docker service scale HWorld_web=3
HWorld_web scaled to 3
overall progress: 0 out of 3 tasks
1/3:
2/3:
3/3:
overall progress: 0 out of 3 tasks
overall progress: 0 out of 3 tasks
```

As you can see, the service is deployed across the nodes in the cluster.



```
ubuntu@ip-172-31-13-58:~$ docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED              STATUS              PORTS                                       NAMES
4c2ad123e315   docker-compose-example_web       "/opt/tomcat/bin/cat…"   About a minute ago   Up About a minute   0.0.0.0:9000->8080/tcp, :::9000->8080/tcp   docker-com
pose-example_web_1
f20763092f14   vniranjan1972/hworld_tomcat:V1   "/opt/tomcat/bin/cat…"   6 minutes ago        Up 6 minutes        8080/tcp                                    HWorld_web
.1.nppzdpfdxacv2a67a9oivycdp
```