



## **SWE4002 - CLOUD COMPUTING**

Project Title : **METRO TICKETING SYSTEM based on  
CLOUD COMPUTING**

**SLOT : A2**

**FACULTY : Dr. Vani M P**

**Jth Component Review**

### **TEAM MEMBERS :**

**BHUVANESH M - 19MIS0066**

**DEEPAN S - 19MIS0102**

**VIPPARTHI VENKATA SAI SRINIDHI - 19MIS0319**

## **ABSTRACT :**

The current metro booking system is human dependent and tedious with regards to the ticket booking process. The core objective of our project is to develop the web-application which will serve as a medium for students/employees/anyone to book the tickets to travel through metro. The principle motive force of this web-application is to ease the process of ticket booking by avoiding the hectic and boisterous process to stand in a queue and book the ticket for the short distance for travelling in metros.

Our application would book the ticket as well as save the travel tickets in the cloud database. Tickets can be bought with the help of smart phone, laptop using the web-application where metro tickets can be carried in user's mobile phone. The ticketing information of the user is securely stored in the cloud database. Additionally, the ticket checker is given the checker application which is utilized for the approval of the ticket appeared by the user. This framework gives the ticket checker web-application to look for the user's ticket with the ticket number or other appropriate information in the cloud database for checking purposes. Consider that the user's display is being damaged and not able to show the ticket due to other reasons like battery failure we have another safeguard alternative to check the ticket by searching in the ticket database with the ticket number or user's other relevant information for validation purpose.

## **KEYWORDS :**

Cloud Computing, Reservation, cloud storage, face recognition

## **INTRODUCTION :**

There has been no advancement in Indian public transport system. It still follows the regular old pattern of ticket booking and checking. With the growing population, the number of travellers ready to travel day by day is expanding abruptly and now the circumstances are deteriorating that individuals don't bother whether they have a ticket or not, they knowingly or sometimes because of some issue they are entered in the train or bus without a ticket. Indian public transport system and IT are loosely bounded. Presently the use of Information Technology is only limited to online checking of schedules and fares of public transport. The main motive of this web-application is to ease the process of ticket booking by avoiding the hectic process to stand in long queues and book the ticket for the short distance travelling in the trains and bus. Users can purchase the ticket over the Internet, 24 hours a day throughout the year, this solves the issue of bus ticket being misplaced or stolen in a real-life scenario. The application may get overloaded due to a huge number of users visiting at once. Thus to solve the issue this system is built up using cloud infrastructure for improved performance.

## **LITERATURE SURVEY :**

A Study by Mohezar et. al. identified trends in e-tickets among urban communities, especially in Kuala Lumpur. This research explores the trends and patterns of use of e-tickets. The study also focused on consumer perspectives for e-tickets in terms of their usability, reliability, protection, convenience and performance. The research also explores the effect of demographic variables on e-ticket acceptance of e-tickets. A survey was conducted amongst Internet users in Kuala Lumpur.

Questionnaires were randomly distributed to 5,000 individuals. Kuala Lumpur was selected to have the largest number of Internet users. The study found that e-tickets are not a new trend, as an almost good number of respondents have been purchasing online tickets for the past two years and the purchase of rail and bus tickets seems to have dominated online ticketing services. It was also found that comfort and ease of use were among the factors that inspired respondents to buy tickets online. The study also found that online ticket purchasers are young, qualified and with a higher income bracket, Sahney et al. found that the modus operandi of the online ticket booking system needs particular attention to factors such as the functionality of online search information, website design, and the capacity of all time network availability for online booking. We propose that the flexibility of the Internet should be combined with the convenience of a simplified decision making and collaborative booking from traditional travel agents. The expertise of travel agents should help online customers to find the best travel option under given constraints and provide efficient support for impulsive decision.

## **PROBLEM STATEMENT :**

The need to build this website was the technological development of almost everything around us. The user needs all the tasks to be accomplished in an effective and relaxed manner. In such a time, there was a desperate need to construct a website for the convenience of the user. Also, this website will aim to solve the tiresome task of managing the crowd easy, without confusion, during ticket booking times. Cloud Technology will help to add flexibility and scalability.

In the past, we were using various documents for user authentication but with time all those cards were replaced by a single point of authentication Aadhar card in India. The main problem with the physical existence of a card is that we may sometimes forget to carry the card and due to carelessness, we may lose it and that can be further used by others for some different works. For travel, we need to carry our ticket as well as a UID card which is always a hectic task for day to day travel. Hence we propose a solution for a system that may take into account the biometrics of a person for authentication.

## **MODULE DESCRIPTION :**

### **Algorithm for Passenger :**

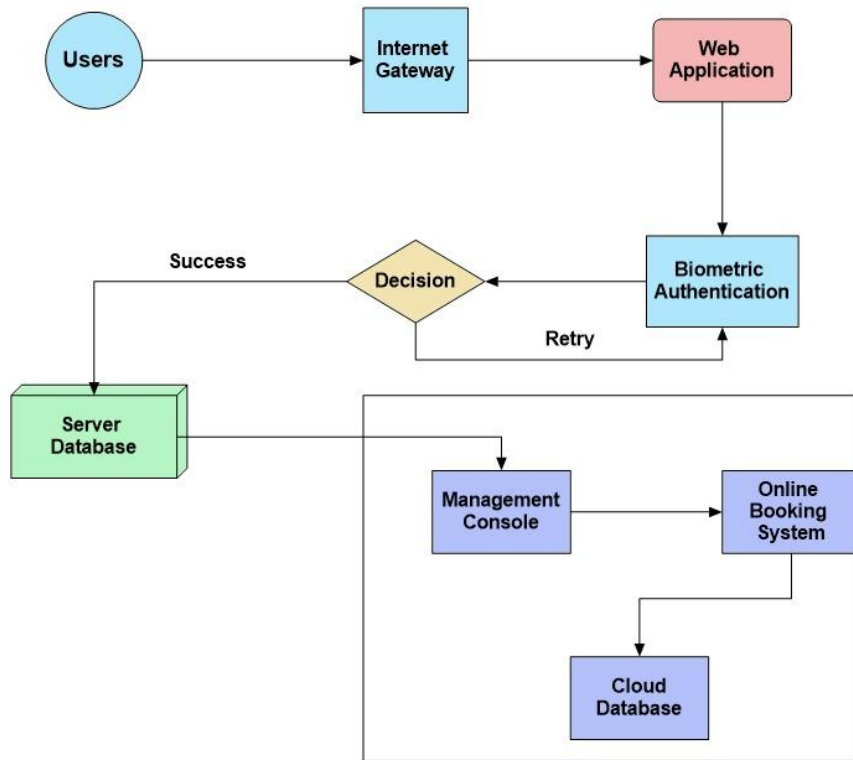
- ✧ Step 1 : Start
- ✧ Step 2 : Input : Login into the passenger login using username and password. System then validates using the data stored in the cloud.
- ✧ Step 3 : Input : If they are new users, they have to fill the new customer form and then they can login into the user login page and these details are stored in the cloud.
- ✧ Step 4 : Output : Passenger can perform the functions such as booking, cancelling, paying, registering the face and verifying tickets.
- ✧ Step 5 : Logout from the portal
- ✧ Step 6 : Stop

### **Algorithm for Admin :**

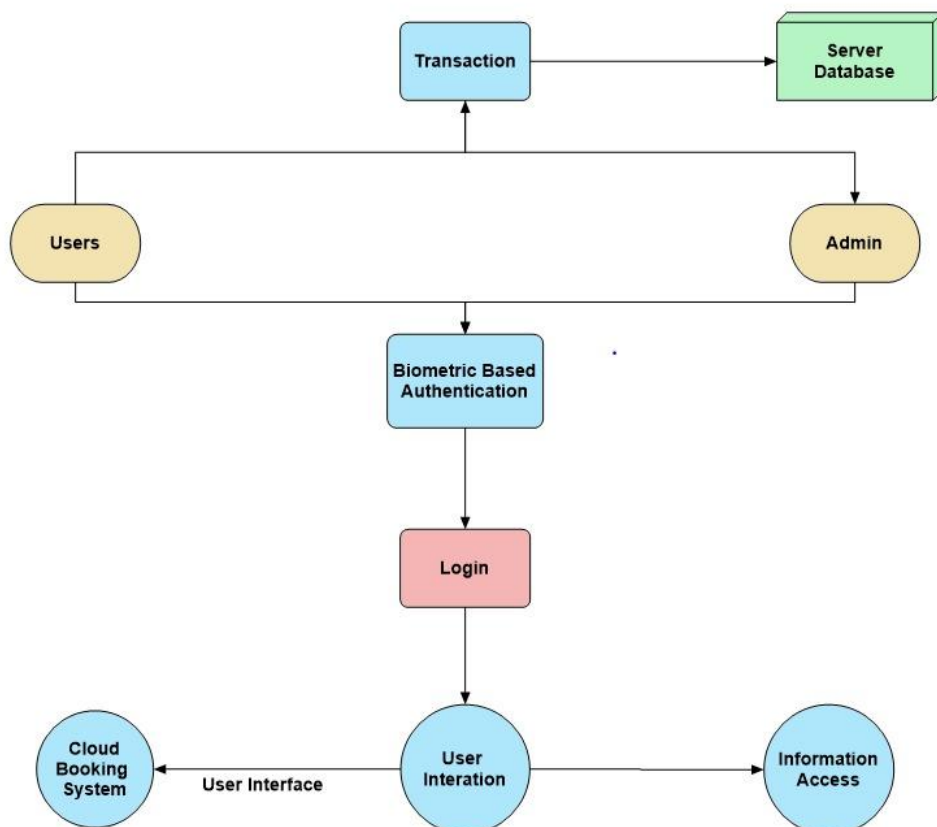
- ✧ Step 1 : Start
- ✧ Step 2 : Input : Login into the employee login using username and password. System then validates using the data stored in the cloud.
- ✧ Step 3 : Output : Admin can perform functions such as create, update and delete the details of the user, tickets booked, seats available and the transaction details of the user in the cloud.
- ✧ Step 4 : Logout from the portal
- ✧ Step 5 : Stop

## ARCHITECTURE :

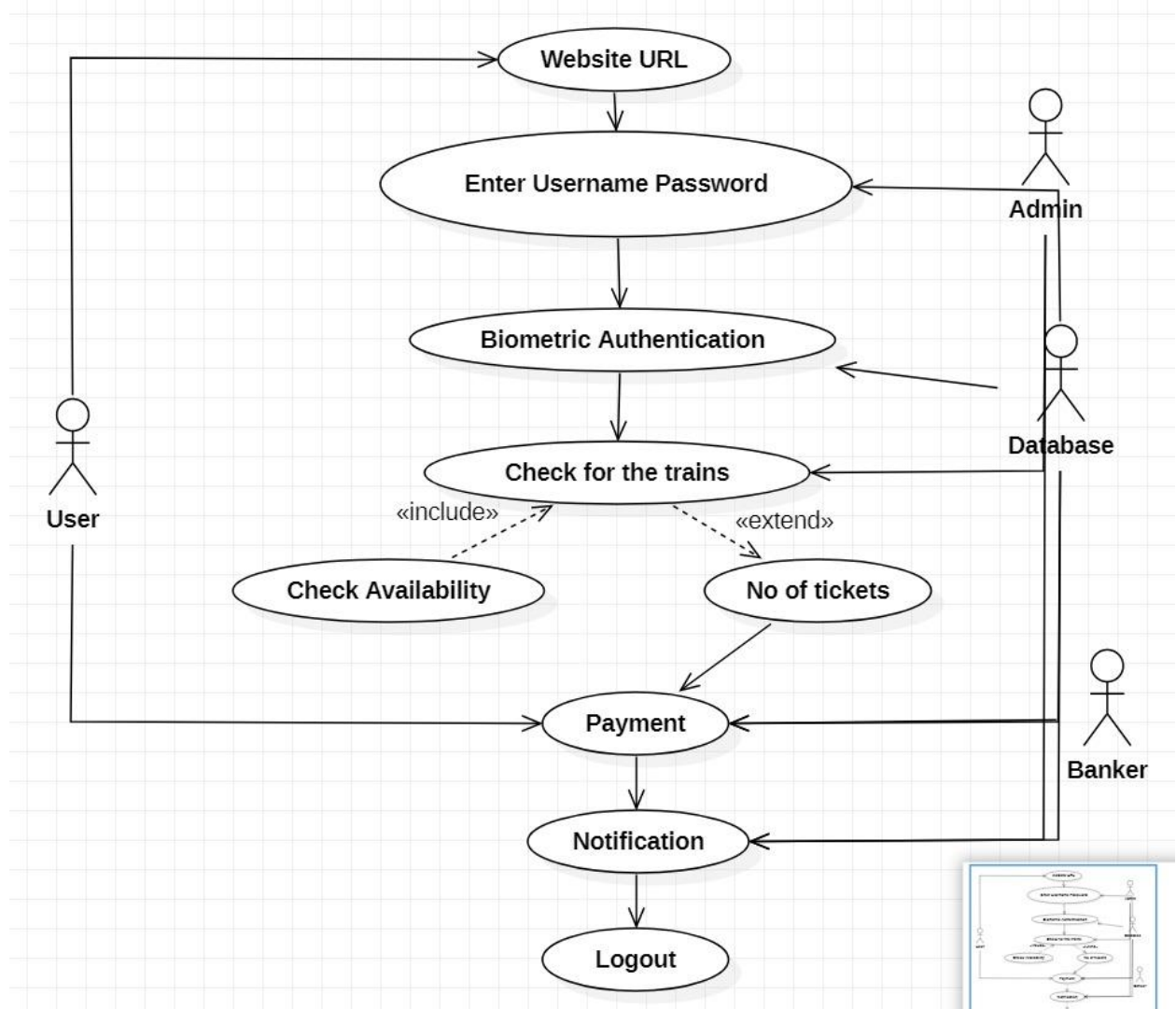
### Low-Level Design :



### High-Level Design :



## USECASE DIAGRAM :



## **TECHNOLOGIES USED :**

### **1) REACT JS**

- React offers some outstanding features that make it the most widely adopted library for frontend app development.
- React is a JavaScript-based frontend library for web development.
- React uses Virtual DOM, thereby creating web applications faster. Virtual DOM compares the components' previous states and updates only the items in the Real DOM that were changed, instead of updating all of the components again, as conventional web applications do.
- Since React employs one-way data binding, all activities stay modular and quick.
- Component Based Architecture simply means that the user interface of an app based on React JS is made up of several components, with each of them having its particular logic, written in JS.
- Due to this, developers can relay the data across the app without the DOM being impacted.
- React JS components play a huge part in deciding the app visuals and interactions.
- The UI of an app is extremely important nowadays; an app with a great UI that offers an exciting browsing and shopping experience to users definitely has much better chances of succeeding than one with a poor interface.
- JSX enables the creation of unique components to suit your requirements; they accept HTML coding and eases the rendering of sub-components.
- The feature of reusing components, enables developers to redistribute the same digital objects. Therefore, even with frequent updates or modification of components that usually impact others, in React it does not become problematic for development.
- The super-fast rendering feature of React brings down the load time of the page drastically, enabling businesses to get their apps ranked on the first page of Google search.

### **2) Mongo DB :**

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

MongoDB supports field, range query, and regular-expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size.

MongoDB scales horizontally using sharding. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more replicas.) Alternatively, the shard key can be hashed to map to a shard – enabling an even data distribution.

### 3) Express Js:

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework –

- Allows to set up middlewares to respond to HTTP Requests.
- Defines a routing table which is used to perform different actions based on HTTP Method and URL.
- Allows to dynamically render HTML Pages based on passing arguments to templates.

### 4) NodeJs:

Node.js is an open-source server environment. Node.js is cross-platform and runs on Windows, Linux, Unix, Mac OS, etc. Node.js is a back-end JavaScript runtime environment. Node.js runs on a JavaScript Engine (i.e. V8 engine) and executes JavaScript code outside a web browser.

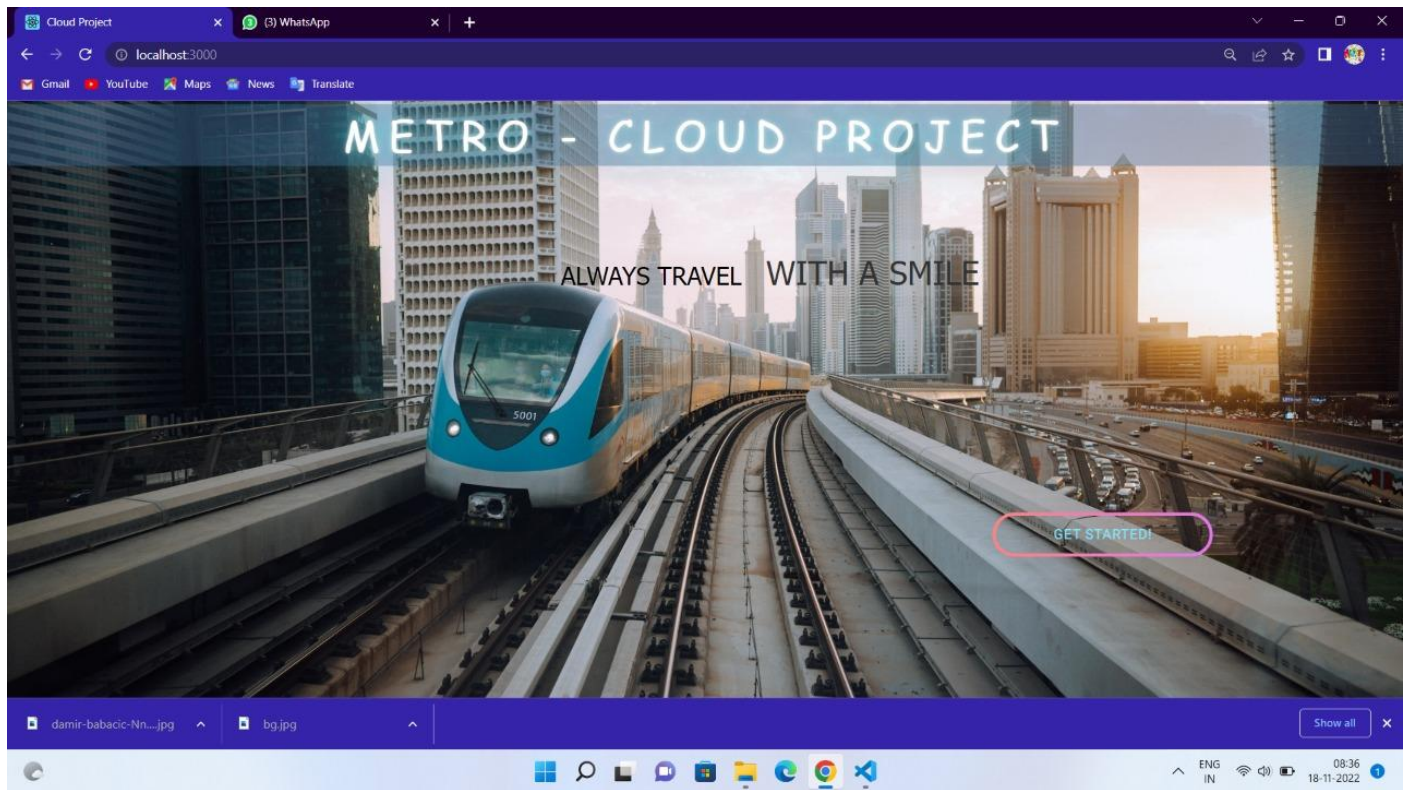
Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications

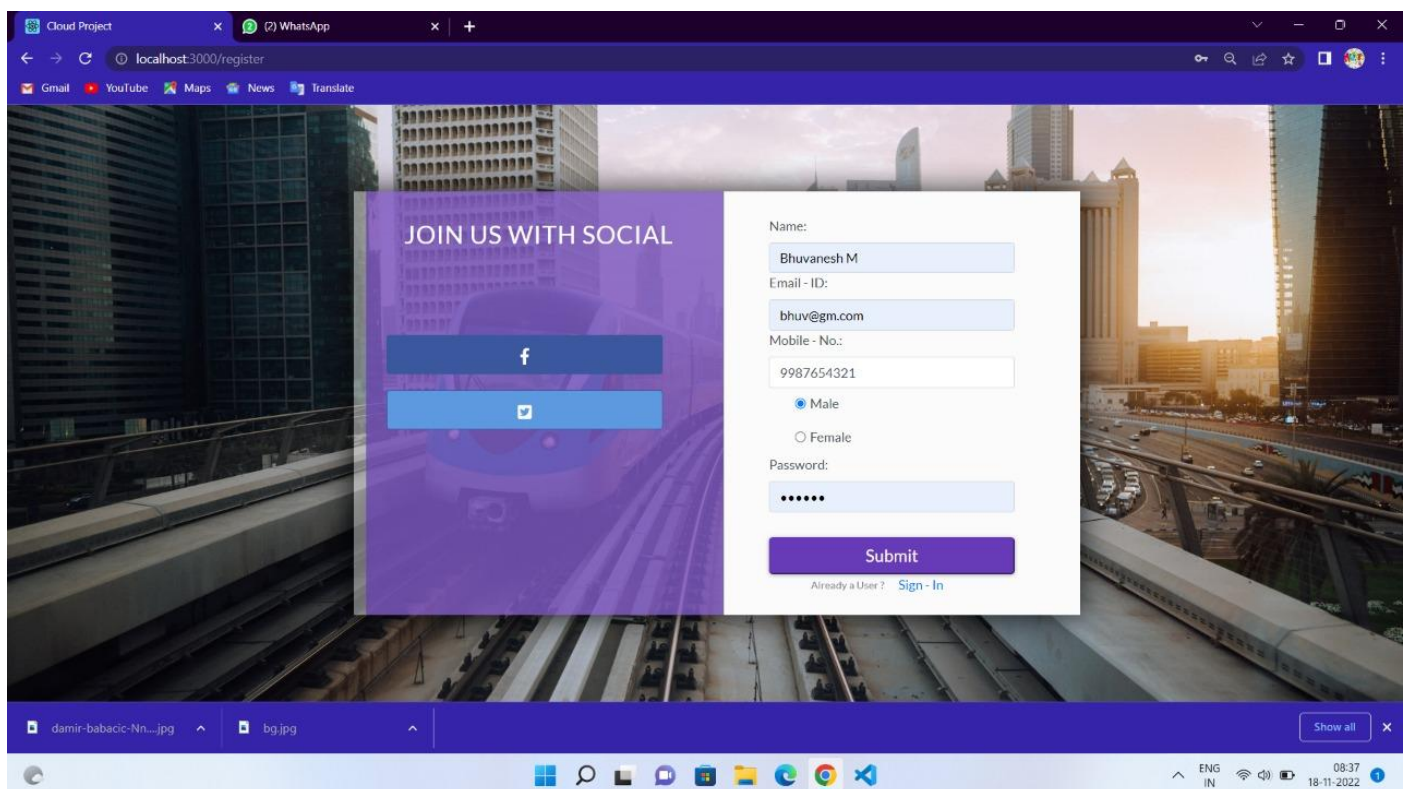


## IMPLEMENTATION SCREENSHOTS :

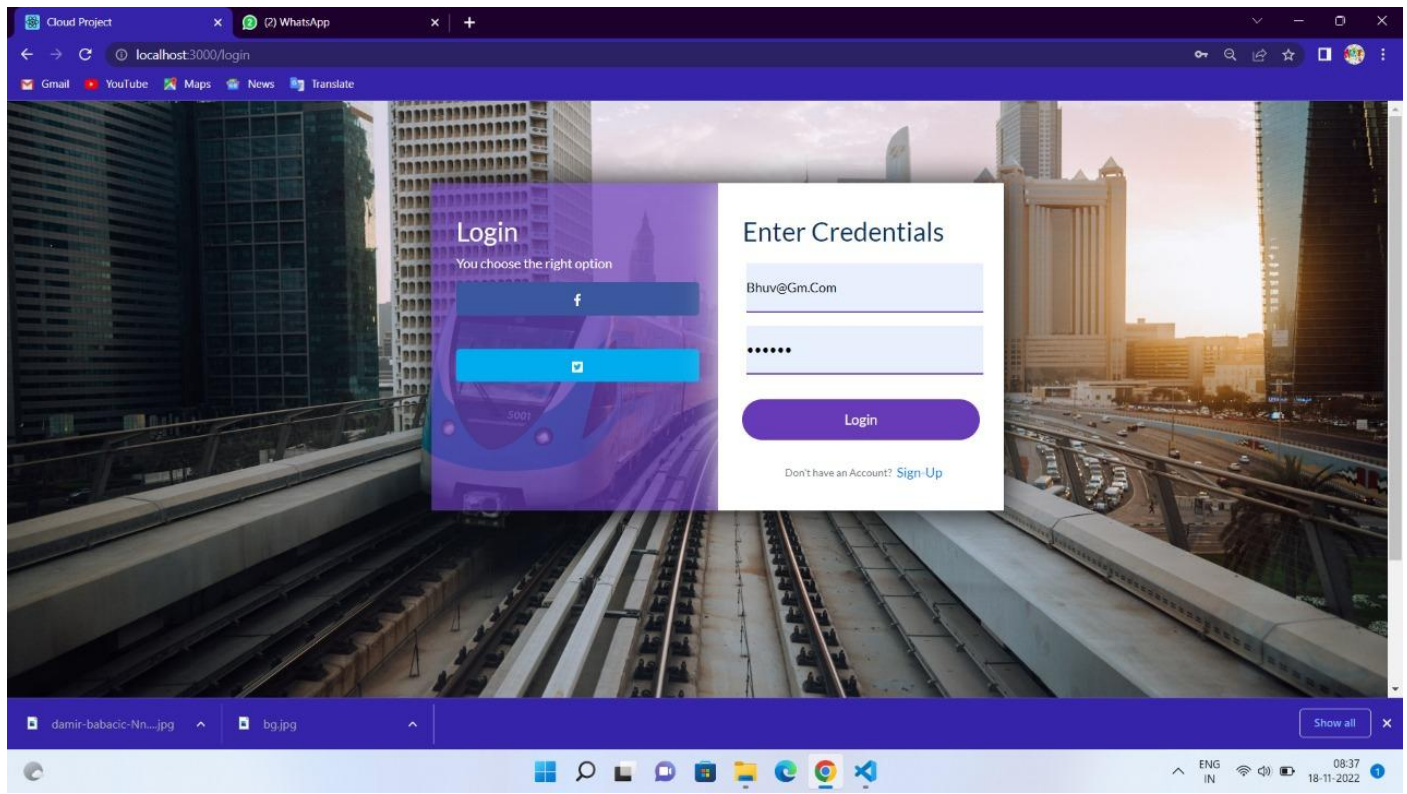
### Home Page :



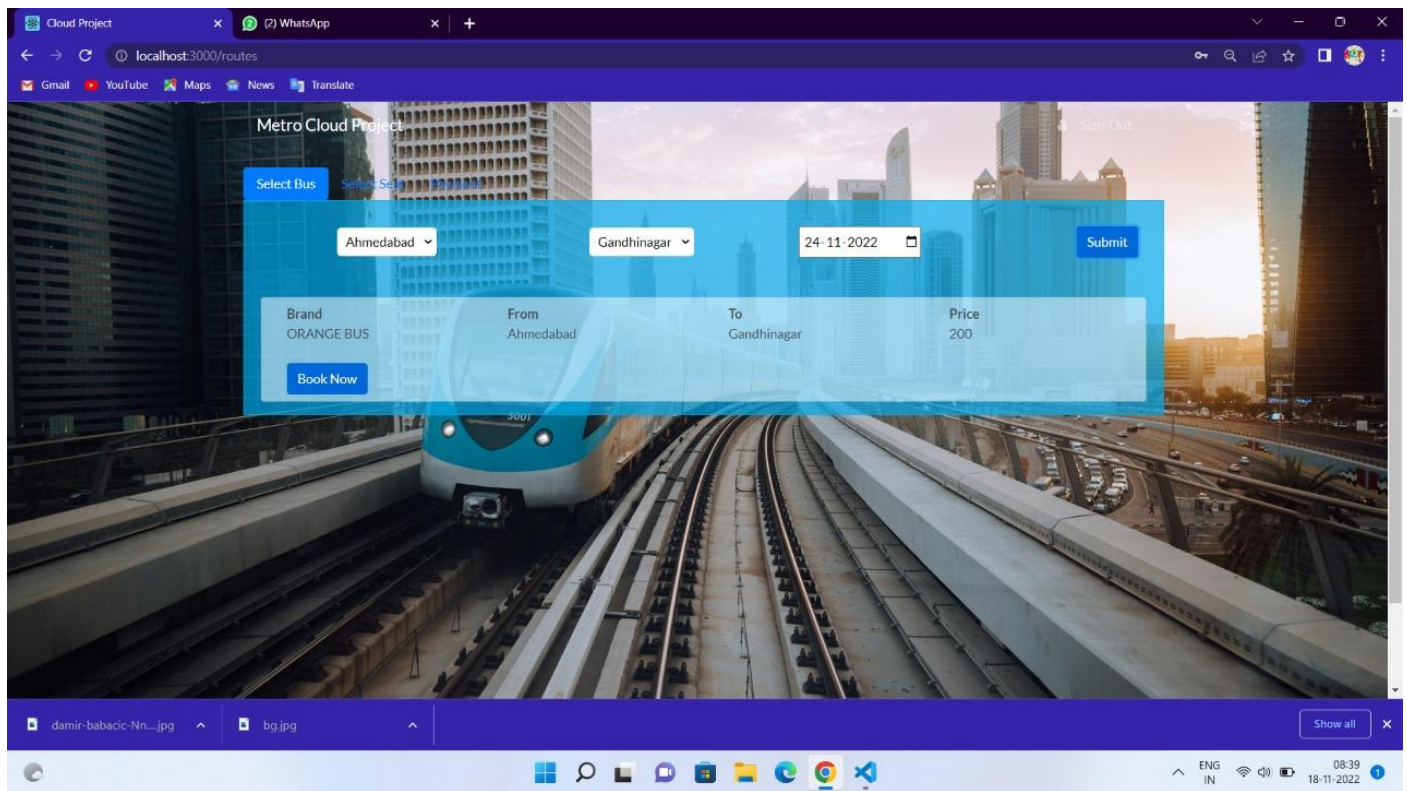
### Signup Page :



## Login Page :

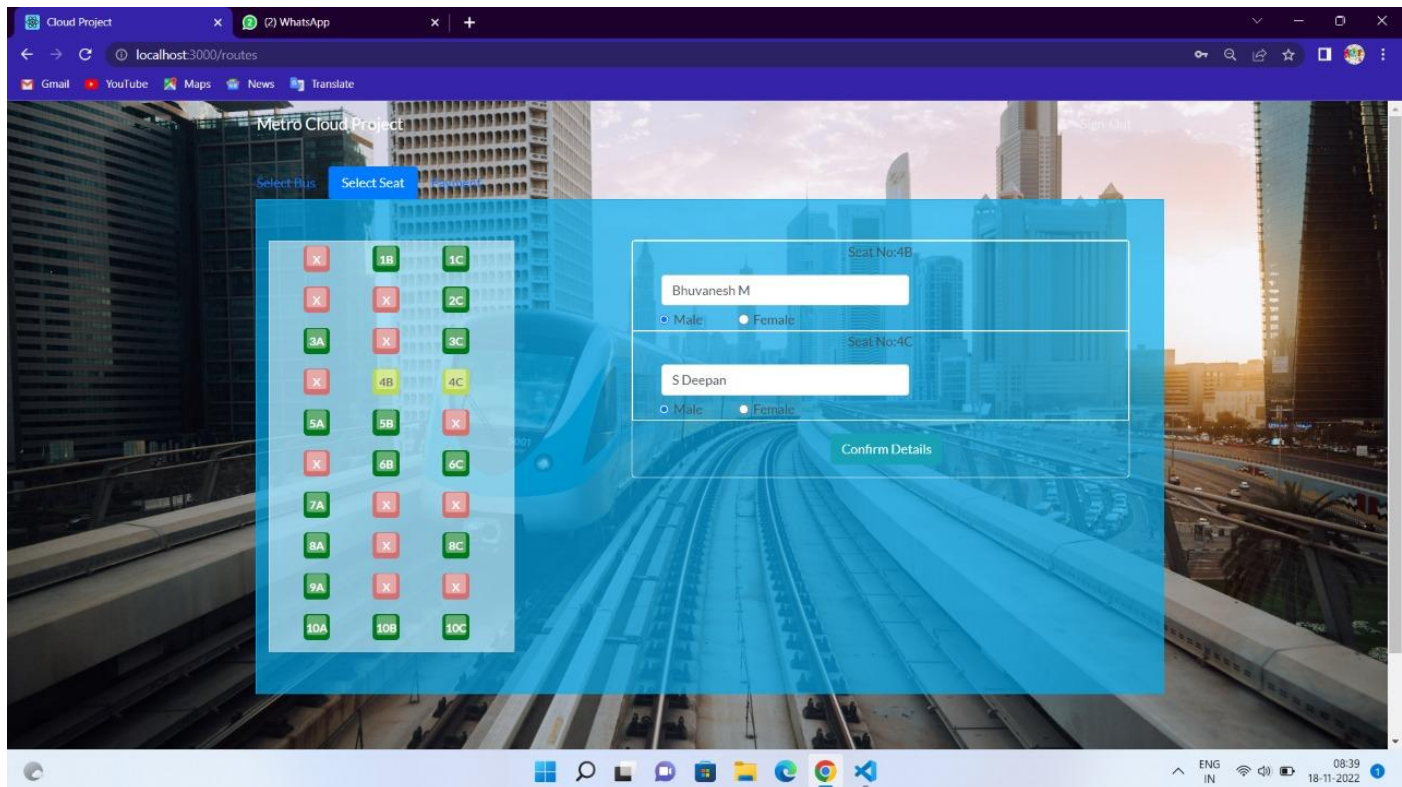


## Search :

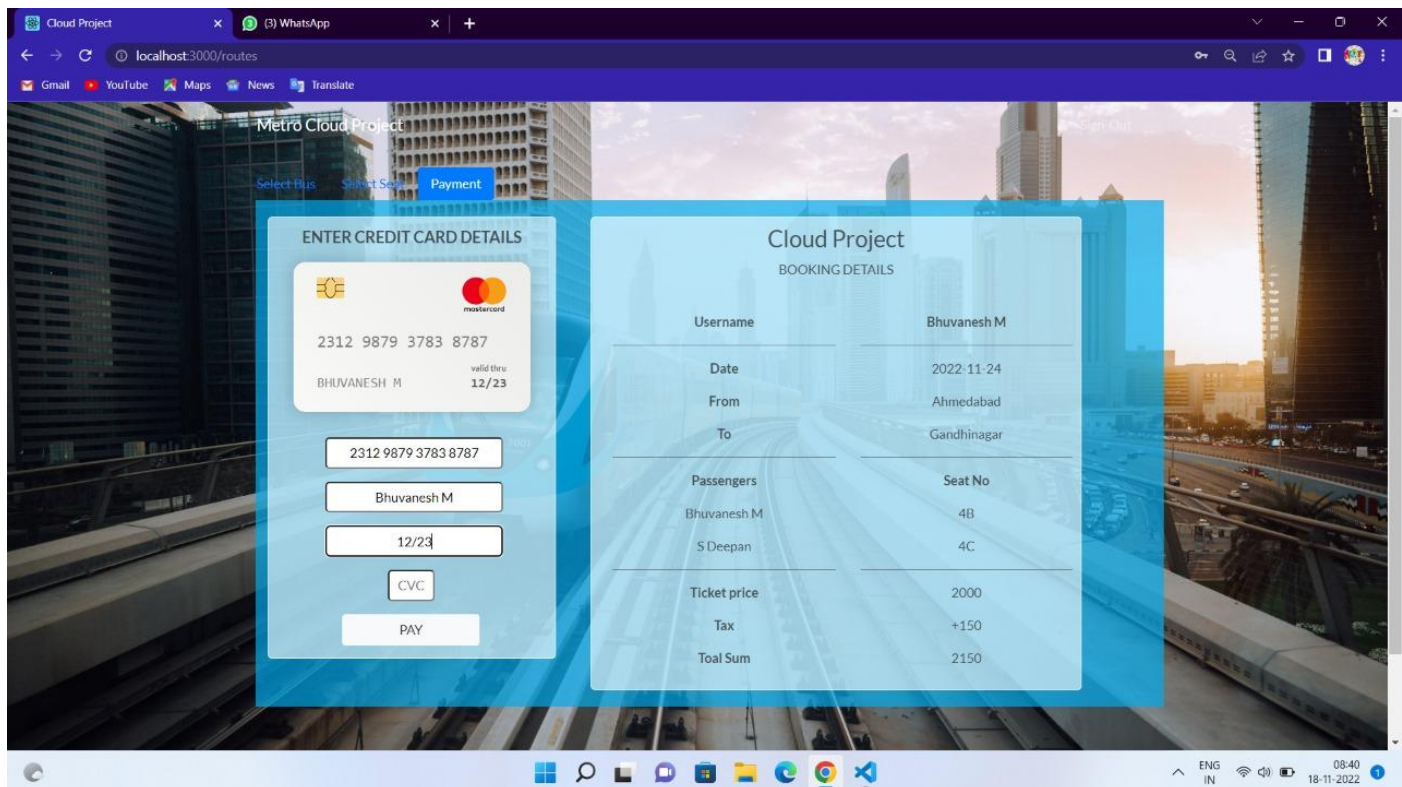


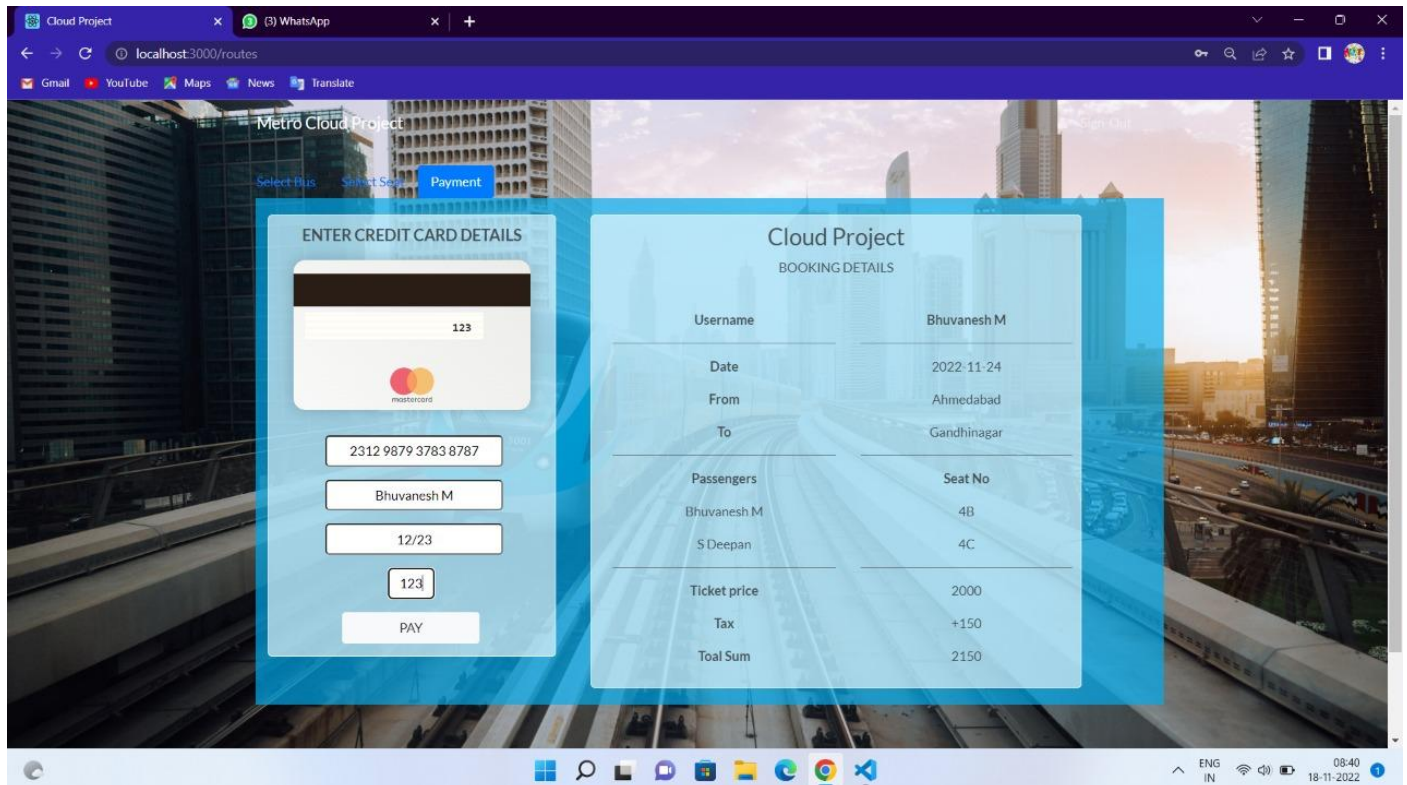


## Seat Selection Page :

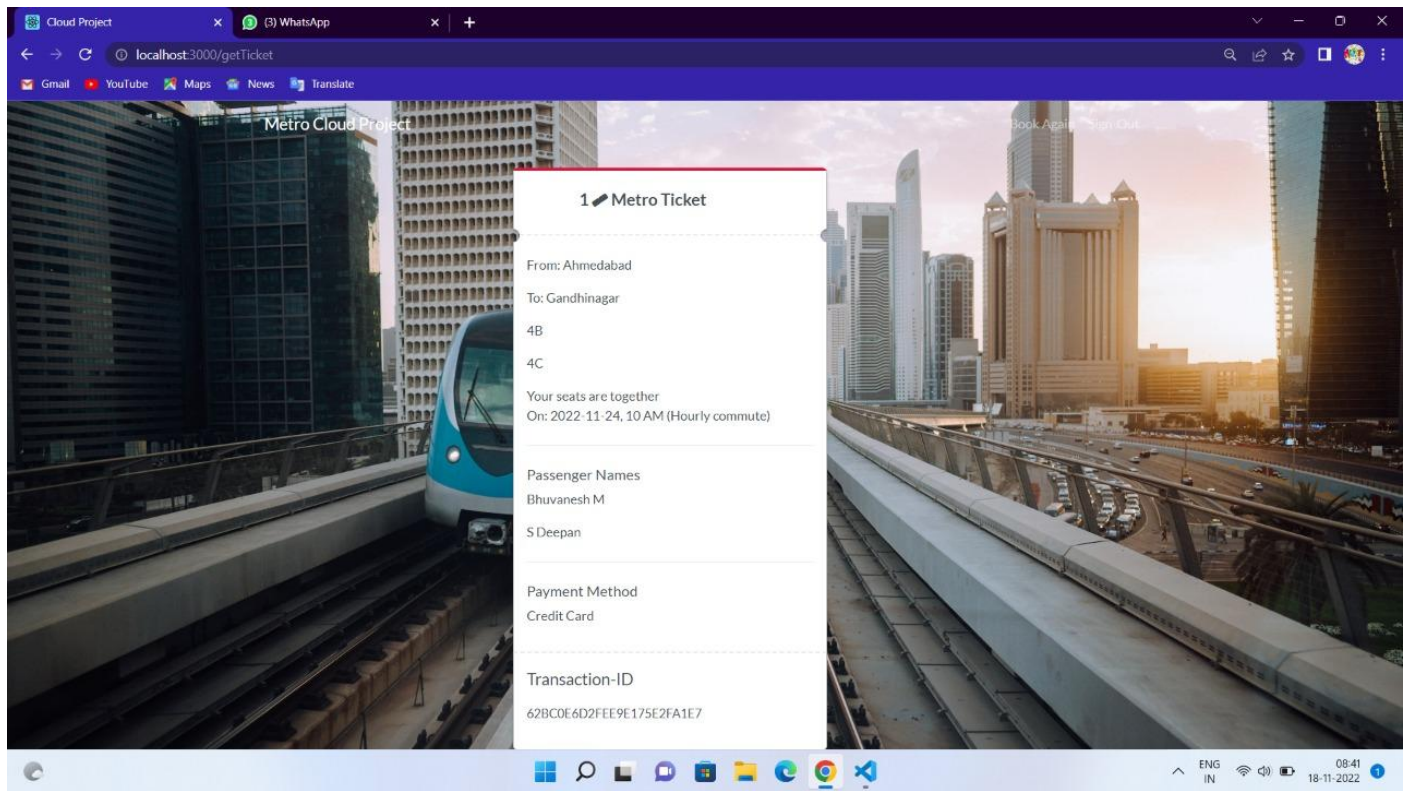


## Payment Screens :





## Ticket :





# MONGODB :

## AWS (Mumbai)

This screenshot shows the MongoDB Atlas 'Cluster0 Overview' page. The browser address bar indicates the URL: `cloud.mongodb.com/v2/63747b70fcd16869053abbed#clusters/detail/Cluster0`. The page header includes navigation links for 'Access Manager', 'Billing', 'All Clusters', and 'Get Help'. The main content area displays the cluster details for 'Cluster0' in the 'AWS Mumbai (ap-south-1)' region, version 5.0.13, with a cluster tier of 'M0 Sandbox (General)'. The 'Overview' tab is active, showing a 'SANDBOX' environment with three nodes: 'ac-...shard-00-00.m...' (SECONDARY), 'ac-...shard-00-01.m...' (PRIMARY), and 'ac-...shard-00-02.m...' (SECONDARY). A central message states 'This is a Shared Tier Cluster' with an 'Upgrade' button. To the right, a graph shows 'Operations R: 0 W: 0' at '0.003/s' for the time period '11/16/22 - 07:57 AM'. Below the graph, 'Logical Size' is shown as '83.7 KB' (512.0 MB max) and 'Connections' as '0' (500 max). A notification bubble indicates 'MongoDB read and write operations on the replica set primary'. The left sidebar contains navigation links for 'DEPLOYMENT', 'Database', 'DATA SERVICES', and 'SECURITY'. The Windows taskbar at the bottom shows the search bar and various application icons.

This screenshot shows the MongoDB Atlas 'Collections' page for the 'test.metro' database. The browser address bar indicates the URL: `cloud.mongodb.com/v2/63747b70fcd16869053abbed#metrics/replicaSet/63747bf3eafa943d70074fe3/explorer/t...`. The page header includes navigation links for 'Access Manager', 'Billing', 'All Clusters', and 'Get Help'. The main content area displays the 'test.metro' database with 'DATABASES: 1' and 'COLLECTIONS: 2'. The 'Collections' tab is active, showing a 'test' namespace with a 'metro' collection. The 'metro' collection details show 'STORAGE SIZE: 36KB', 'LOGICAL DATA SIZE: 5.83KB', 'TOTAL DOCUMENTS: 24', and 'INDEXES TOTAL SIZE: 36KB'. The 'Find' tab is selected, showing a filter bar with the query `{ field: 'value' }` and an 'Apply' button. Below the filter bar, a document is displayed with the following fields: `_id: ObjectId('63747ef412d73c4e8fa228bf')`, `CompanyName: "RED BUS"`, `busType: "Multi AXLE AC"`, `busNumber: "TN 27 110099"`, `startCity: "Chennai"`, and `destination: "bangalore"`. The left sidebar contains navigation links for 'DEPLOYMENT', 'Database', 'DATA SERVICES', and 'SECURITY'. The Windows taskbar at the bottom shows the search bar and various application icons.

Open Access proceedings Jour... Automated\_Market\_Basket\_An... Market basket analysis visualiz... Projects | Cloud: MongoDB Cl... Data | Cloud: MongoDB Cl...

cloud.mongodb.com/v2/63747b70fcd16869053abbed#metrics/replicaSet/63747b73eafa943d70074fe3/explorer/t...

S's Org - 202... Access Manager Billing All Clusters Get Help S

cloud Atlas App Services Charts

DEPLOYMENT Database Data Lake PREVIEW DATA SERVICES Triggers Data API Data Federation SECURITY Database Access Network Access Advanced

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive

DATABASES: 1 COLLECTIONS: 2 VISUALIZE YOUR DATA REFRESH

+ Create Database

Search Namespaces

test metro users

test.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 5.9KB TOTAL DOCUMENTS: 26 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: 'value' } OPTIONS Apply Reset

QUERY RESULTS: 1-20 OF MANY

```
{
  "_id": ObjectId('63747cae6e5bec8918687cbd'),
  "name": "bhuv",
  "email": "bhuv@gm.com",
  "password": "$2b$10$Vtuz4Gy.wZV5LkespnldEe/juzP3vW.FbgDw5dxQisrBz3FtpXrCa"
```

Type here to search

ENG IN 06:49 PM 16/11/2022

Open Access proceedings Jour... Automated\_Market\_Basket\_An... Market basket analysis visualiz... Projects | Cloud: MongoDB Cl... Metrics | Cloud: MongoDB Cl...

cloud.mongodb.com/v2/63747b70fcd16869053abbed#host/replicaSet/63747b73eafa943d70074fe3

S's Org - 202... Access Manager Billing All Clusters Get Help S

cloud Atlas App Services Charts

DEPLOYMENT Database Data Lake PREVIEW DATA SERVICES Triggers Data API Data Federation SECURITY Database Access Network Access Advanced New On Atlas 2

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive

GRANULARITY Auto ZOOM 1 hour CURRENT DISPLAY 11/16/2022 12:20pm to 11/16/2022 01:20pm AT 1 MINUTE GRANULARITY

TOGGLE MEMBERS [S] [S] [P] [S] ADD CHART [ ] DISPLAY OPCOUNTERS ON SEPARATE CHARTS [X] DISPLAY TIMELINE ANNOTATIONS

ac-lquondu-shard-00-00.mlvnah2.mong... :27017 ac-lquondu-shard-00-01.mlvnah2.mong... :27017 ac-lquondu-shard-00-02.mlvnah2.mong... :27017

Opcounters

1.5/s 1/s 0.5/s 0/s

2022/11/16 12:51:50: COMMAND: 0/S QUERY: 0/S UPDATE: 0/S DELETE: 0/S GETMORE: 0/S INSERT: 0/S

2022/11/16 12:51:50: COMMAND: 0/S QUERY: 0/S UPDATE: 0/S DELETE: 0/S GETMORE: 0/S INSERT: 0/S

2022/11/16 12:51:50: COMMAND: 0/S QUERY: 0/S UPDATE: 0/S DELETE: 0/S GETMORE: 0/S INSERT: 0/S

Type here to search

ENG IN 06:50 PM 16/11/2022

## **CONCLUSION & FUTURE WORK :**

Using this system we eradicated the hassles of day to day traveling. There is now no need to carry physical tickets/tokens or any other UID card/documents for the sake of traveling. With this proposed methodology, the user will be ensured a more comfortable and convenient travel experience.

Use of ECC (Eliptic curve cryptography) for securing sensitive facial data, since ECC is perfect for securing data generated by low powered devices such as a IoT devices with a camera. As mentioned IoT devices can be used instead of a full-fledged system. A mobile app can be developed which eliminates the need for a separate ticketing section-authentication.