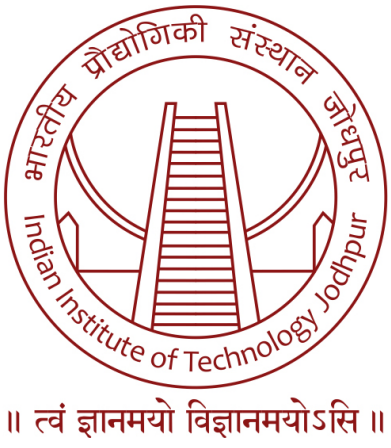
Design Documentation Topic: IITJ Travel Sathi - A Platform to Find Companions for Shared Rides

A Design Documentation Submitted by

Akaash Chatterjee - M24CSE002

Aman Saini - M24CSE003

Bera Swaminath Ansuman Sabita - M24CSE007



Indian Institute of Technology Jodhpur

Department of Computer Science and Engineering

September, 2024

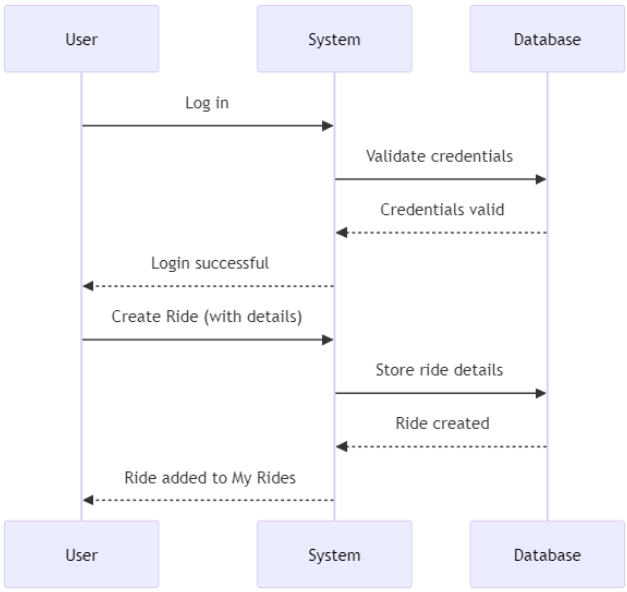
Contents

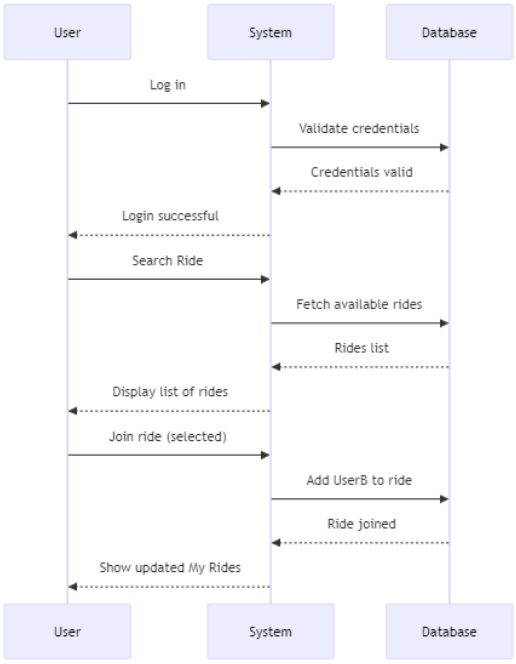
1 Objective

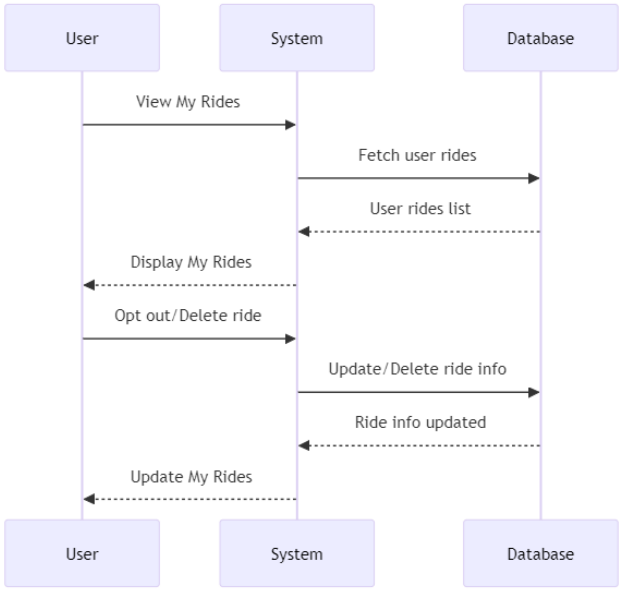
The project aims to create a secure online platform that facilitates shared transportation for college students, starting with rides between key locations like IIT Jodhpur and Jodhpur Junction. Users can post when they are looking for companions for a ride, and others can search for available rides to join. To ensure privacy and safety, the platform offers gender-specific pairing options—female users can choose to travel with female partners only, and the same applies to male users, with an additional co-ed option available. No personal details are shared among users. The platform will also include a database of trusted autorickshaw drivers operating within the IITJ campus. The final aim is to apply containerization using Docker and virtualization based on a Fedora VM using VirtualBox and compare the performance metrics of our website on both.

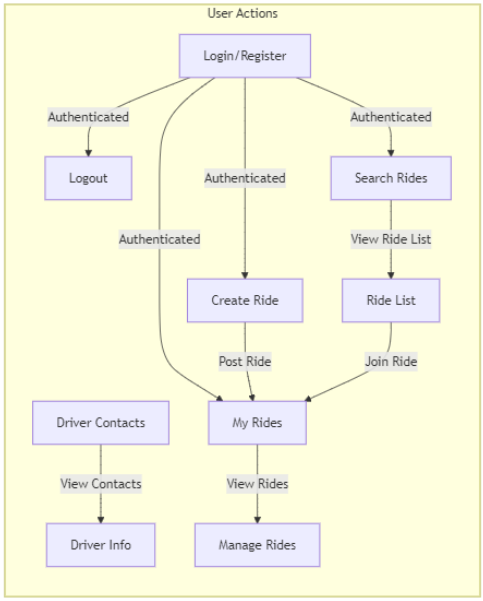
2 System Specifications

Architecture: We have used a *monolithic architecture pattern* for our project considering the initial requirements and constraints.




User
System
Database
Log in
Validate credentials
Credentials valid
Login successful
Search Ride
Fetch available rides
Rides list
Display list of rides
Join ride (selected)
Add UserB to ride
Ride joined
Show updated My Rides
User
System
Database


User
System
Database
View My Rides
Fetch user rides
User rides list
Display My Rides
Opt out/Delete ride
Update/Delete ride info
Ride info updated
Update My Rides
User
System
Database



Technologies used:

- Frontend : HTML,CSS,Javascript and Bootstrap
- Backend : Flask
- Database: SQLite3

3 Functional Requirements:

1. **User Registration**: Users can simply register with a valid email and password and no other detail,ensuring full privacy.We use bcrypt for password hashing and secured storage
2. **User Login**: Users ,if already registered, can log into their accounts with valid email and password.Without logging in,one cannot access the features of the website
3. **Create a ride:** Users can create a ride of their choice by simply going to the create a ride page and entering the appropriate details such as starting point,ending point,meeting point,departure time,cost per head,available seats and preferred gender of companions
4. **Find a ride:** Users can find appropriate rides that they can join by going to the find rides page.It will show all active rides at that moment.
5. **View your rides:** Users can see the rides they have joined or created under the MyRides page
6. **Join a ride:** Users can join an appropriate ride by clicking on Join a ride on eligible rides.Male users can join only male and coed rides while female can join female and coed rides respectively
7. **Opt out from a ride:** Users can opt out from a upcoming ride by clicking on opt out from their upcoming rides under the my rides page

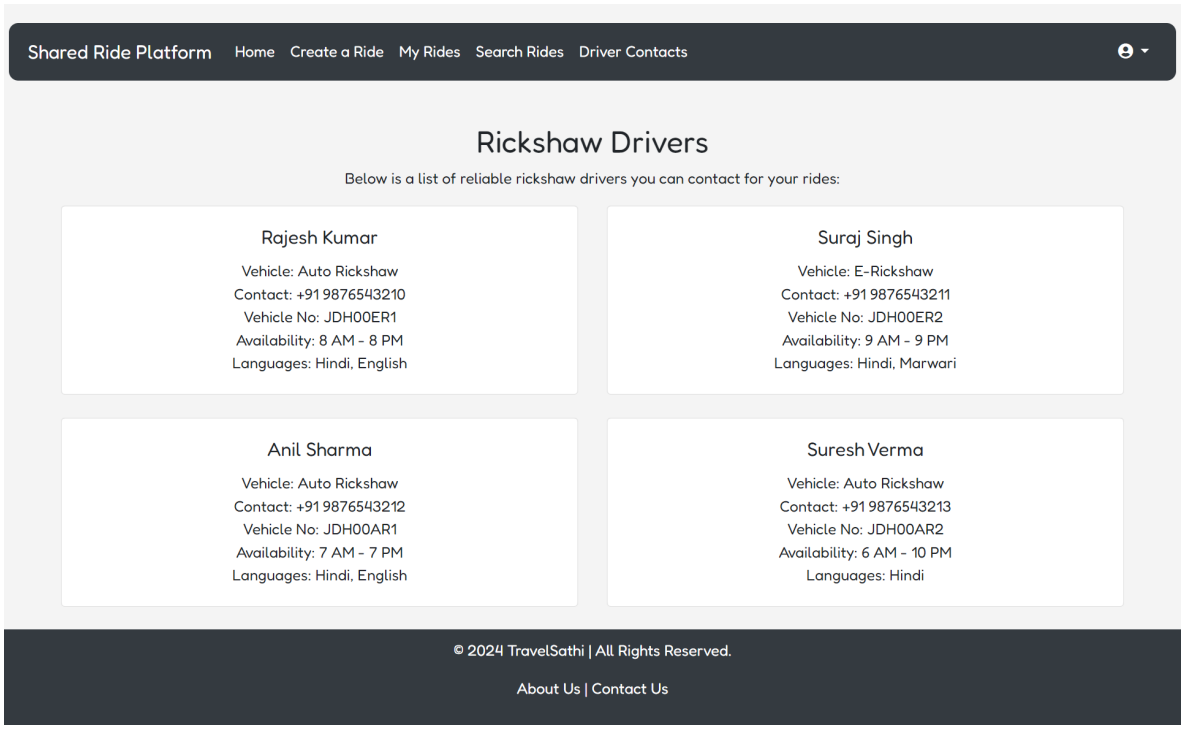
8. **Delete a ride:** Users can delete a ride that they had created in case they wish to cancel the event.However, other users cannot delete rides that weren't created by them.
9. **About Us:** Users can find the details about the app by going to the about us page.
10. **Contact Us:** Users can find our contact information by going to the contact us page.

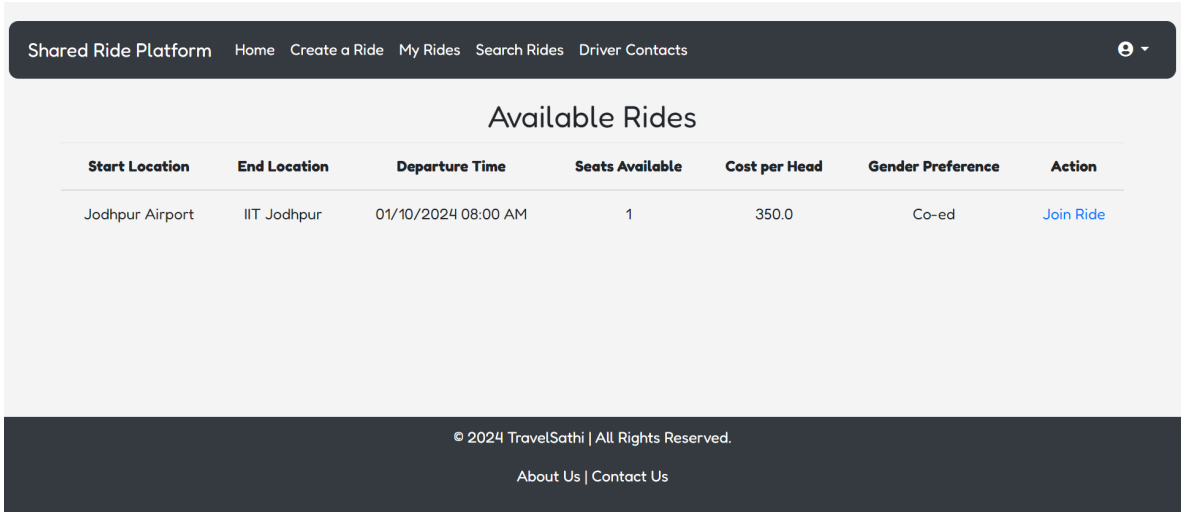
4 UI Design:

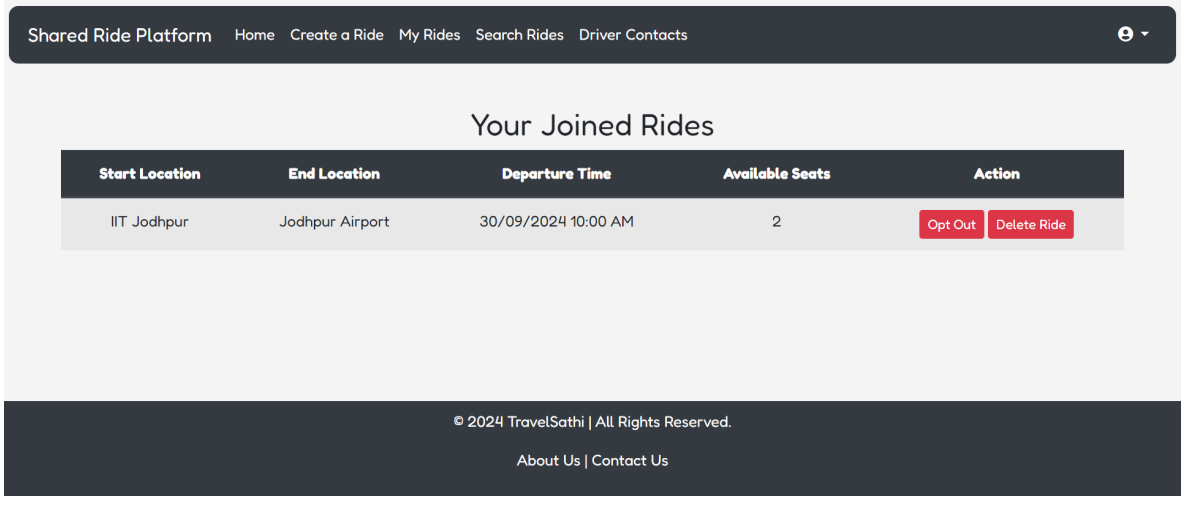
4.1 Structure of the HTML files:-

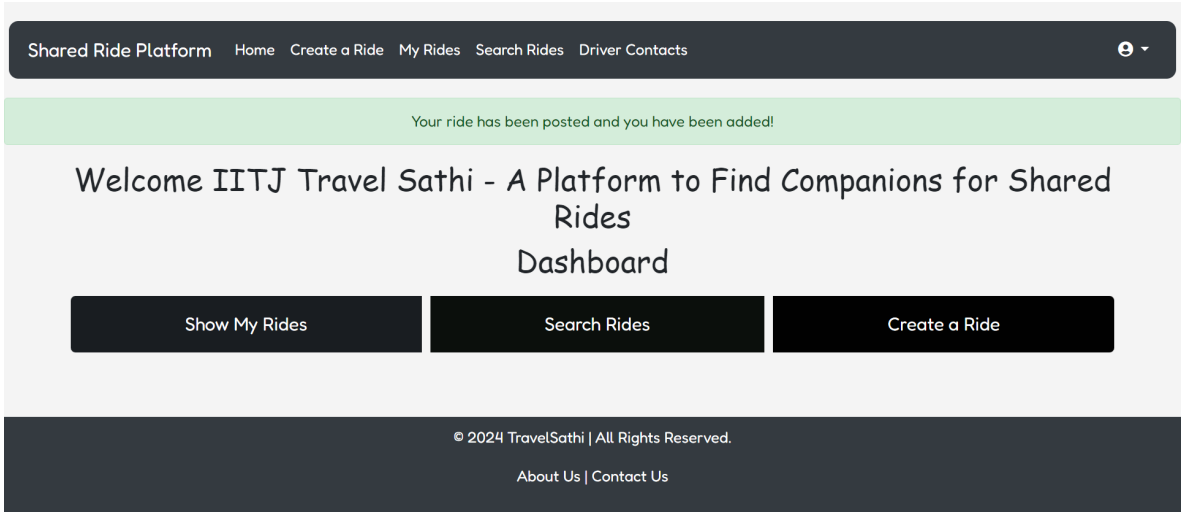
1. **base.html**: This file serves as the main template from which other templates extend.It contains the code for Navigation bar and Footer.
2. **index.html**:This is the main landing page of the platform.For logged in users,it shows buttons to redirect to creating,searching or viewing currently joined rides.For non authenticated users,it shows a message to login or signup .
3. **create_ride.html**:Helps us create new rides.We can specify starting point,ending point,meetup location,departure time,no of passengers,cost per head, and nature of ride(male only,female only,coed)
4. **joined_rides.html**:Displays the rides that the user has joined.User can delete or opt out from rides here.
5. **search_rides.html**:Displays all currently available rides for a user to book.Only the rides that meet the gender specific criteria are allowed.
6. **drivers.html**:Lists the available drivers inside campus and their details.
7. **register.html**:Allows the user to register using name,email,password and gender.
8. **login.html**:Allows the user to login using their email and password
9. **about.html**: Displays information about the dev team
10. **contact.html**: Contact information to reach out to our team

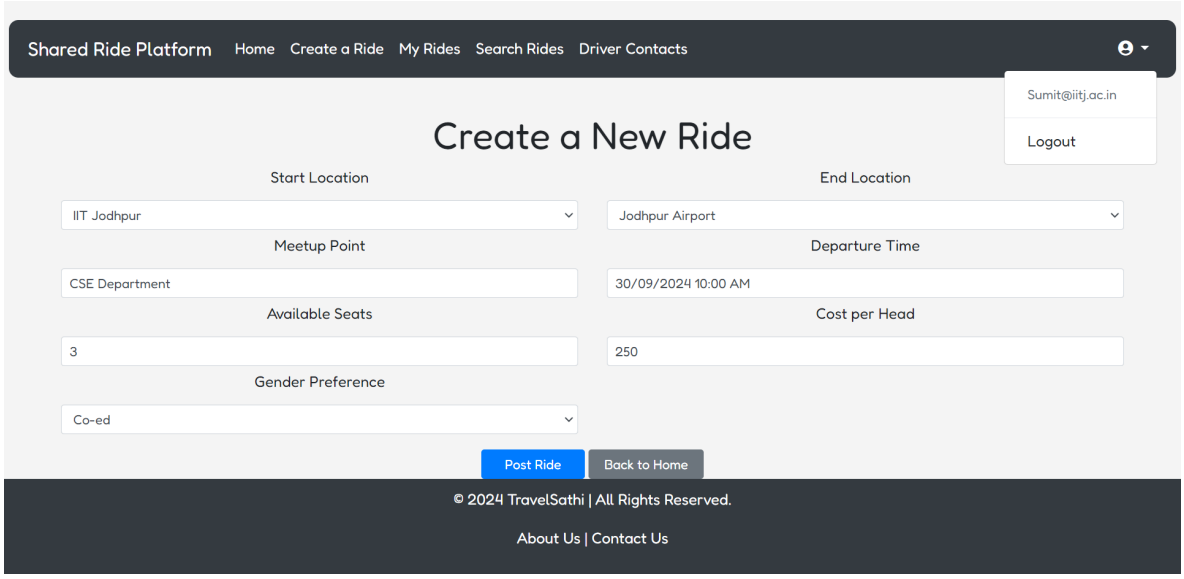
4.2 Screenshots of the pages:-

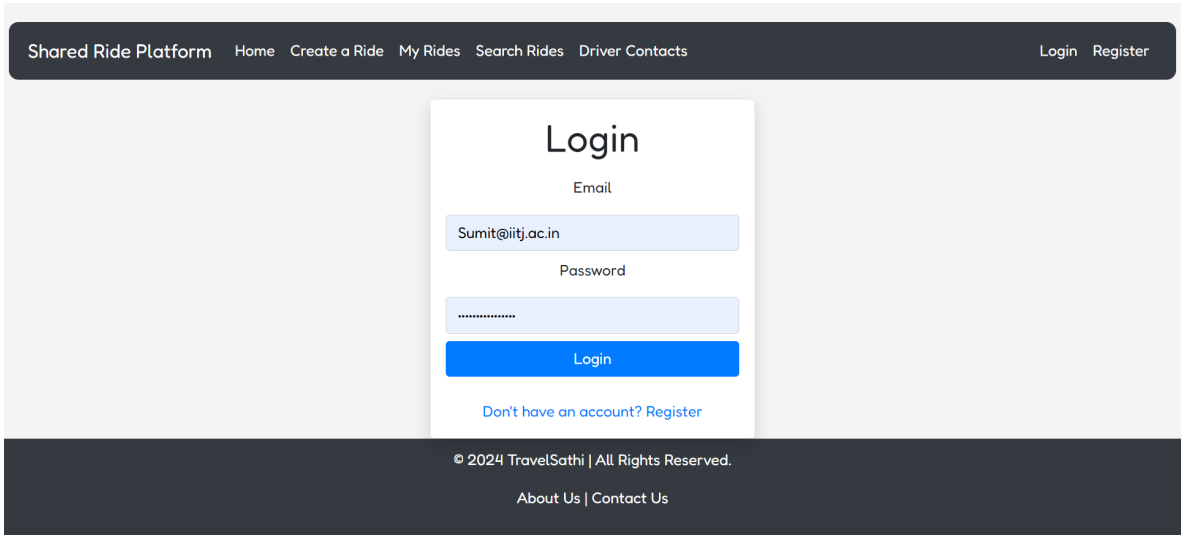


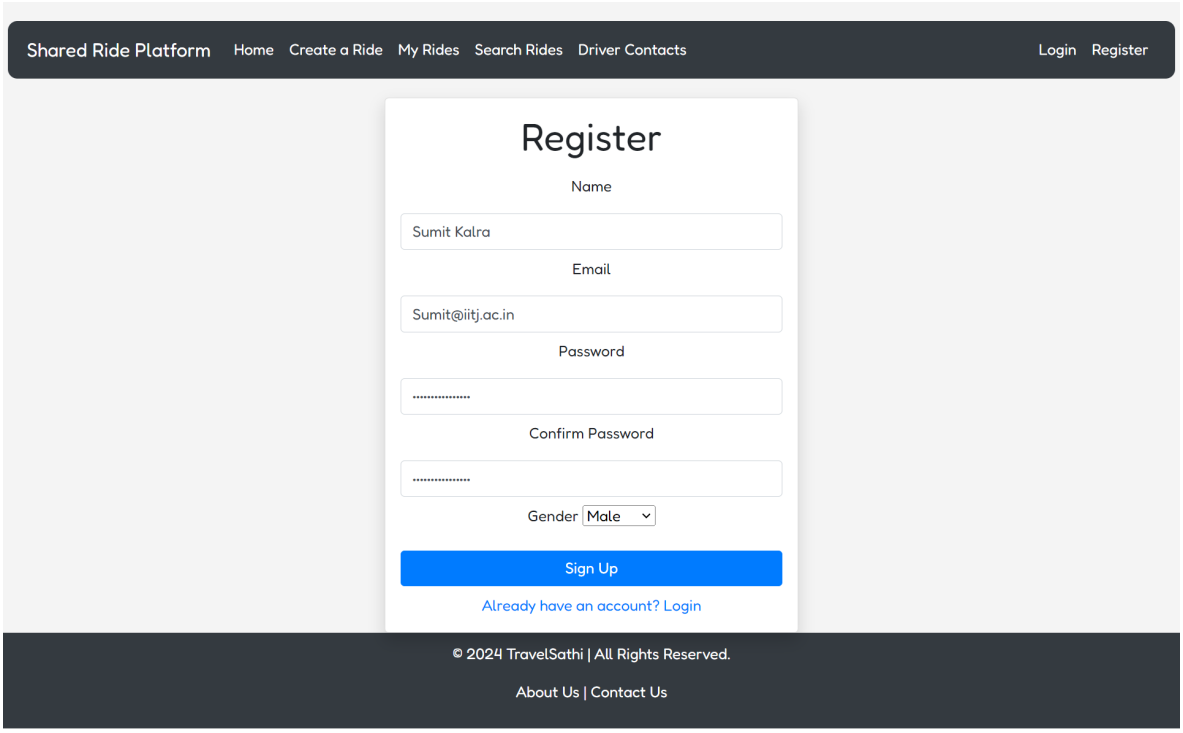


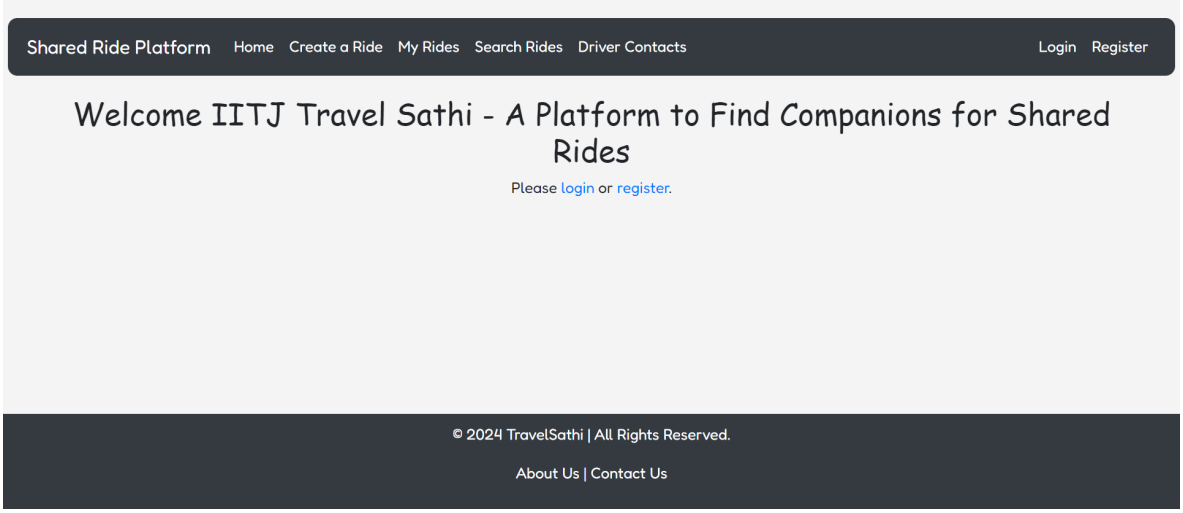



Shared Ride Platform
Home
Create a Ride
My Rides
Search Rides
Driver Contacts
Your ride has been posted and you have been added!
Welcome IITJ Travel Sathi - A Platform to Find Companions for Shared Rides
Dashboard
Show My Rides
Search Rides
Create a Ride
© 2024 TravelSathi | All Rights Reserved.
About Us | Contact Us


Shared Ride Platform
Home
Create a Ride
My Rides
Search Rides
Driver Contacts
Sumit@iitj.ac.in
Logout
Create a New Ride
Start Location
IIT Jodhpur
End Location
Jodhpur Airport
Meetup Point
CSE Department
Departure Time
30/09/2024 10:00 AM
Available Seats
3
Cost per Head
250
Gender Preference
Co-ed
Post Ride
Back to Home
© 2024 TravelSathi | All Rights Reserved.
About Us | Contact Us


Shared Ride Platform
Home
Create a Ride
My Rides
Search Rides
Driver Contacts
Login
Register
Login
Email
Sumit@iitj.ac.in
Password
Login
Don't have an account? Register
© 2024 TravelSathi | All Rights Reserved.
About Us | Contact Us


Shared Ride Platform
Home
Create a Ride
My Rides
Search Rides
Driver Contacts
Login
Register
Register
Name
Sumit Kalra
Email
Sumit@iitj.ac.in
Password
Confirm Password
Gender
Male
Sign Up
Already have an account? Login
© 2024 TravelSathi | All Rights Reserved.
About Us | Contact Us


Shared Ride Platform
Home
Create a Ride
My Rides
Search Rides
Driver Contacts
Login
Register
Welcome IITJ Travel Sathi - A Platform to Find Companions for Shared Rides
Please login or register.
© 2024 TravelSathi | All Rights Reserved.
About Us | Contact Us

5 Database Design:

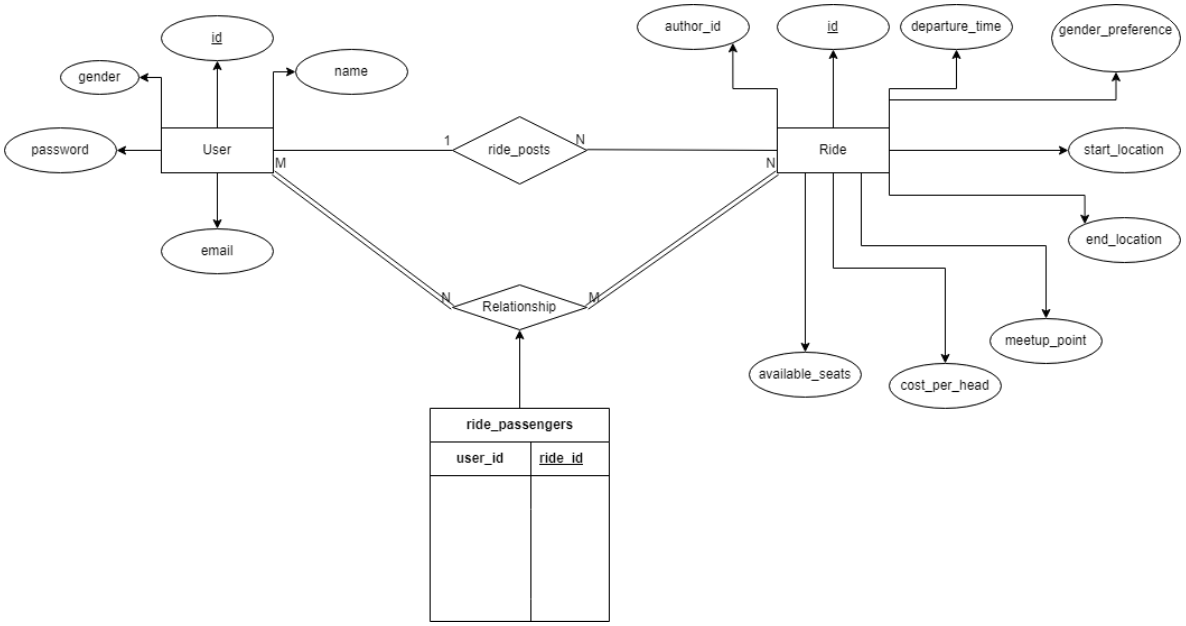


Table : User

Attributes:

- **id**: This is the primary key
- **name**: User's name
- **email**: User's email
- **password**: User's password
- **gender**: User's gender(male/female)

Table :Ride

Attributes:

- **id**: Primary key
- **start_location**: The starting point of the ride
- **end_location**: The destination point of the ride
- **meetup_point**: The location where joined users should meet for the ride
- **departure_time**: Time denoting commencement of the trip
- **available_seats**: Number of seats that are available in the ride
- **cost_per_head**: The cost per head for the trip
- **gender_preference**: Whether the ride has gender restrictions
- **author_id**: Foreign key that links to the User who created the ride

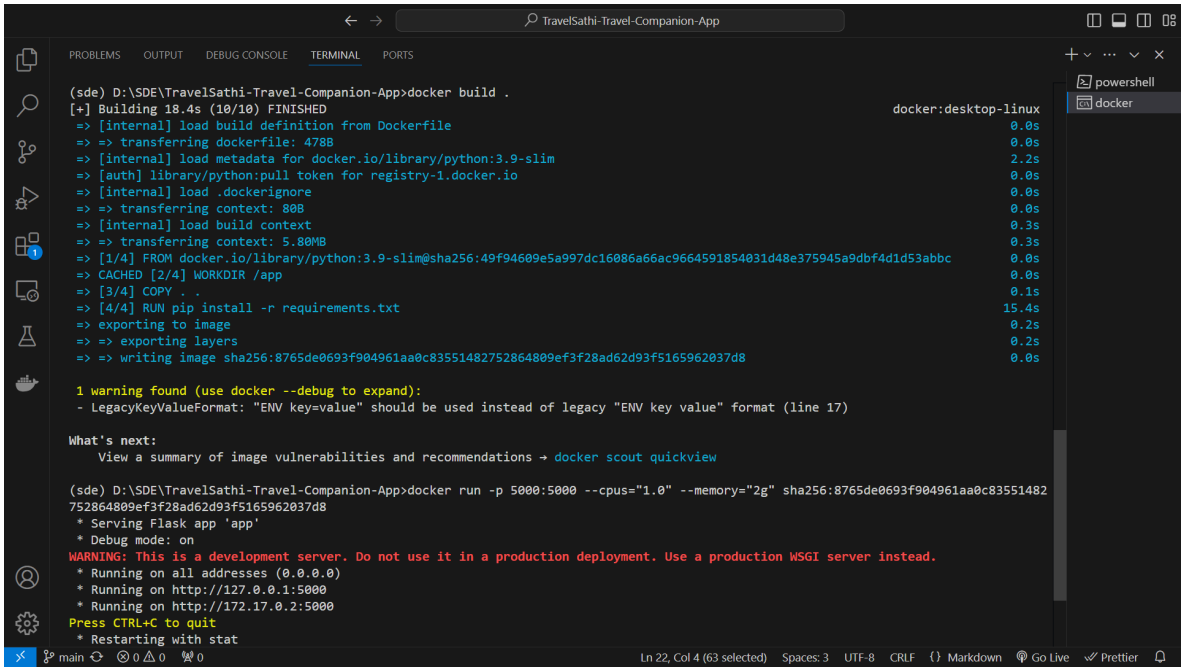
6 Analysis of Routes:

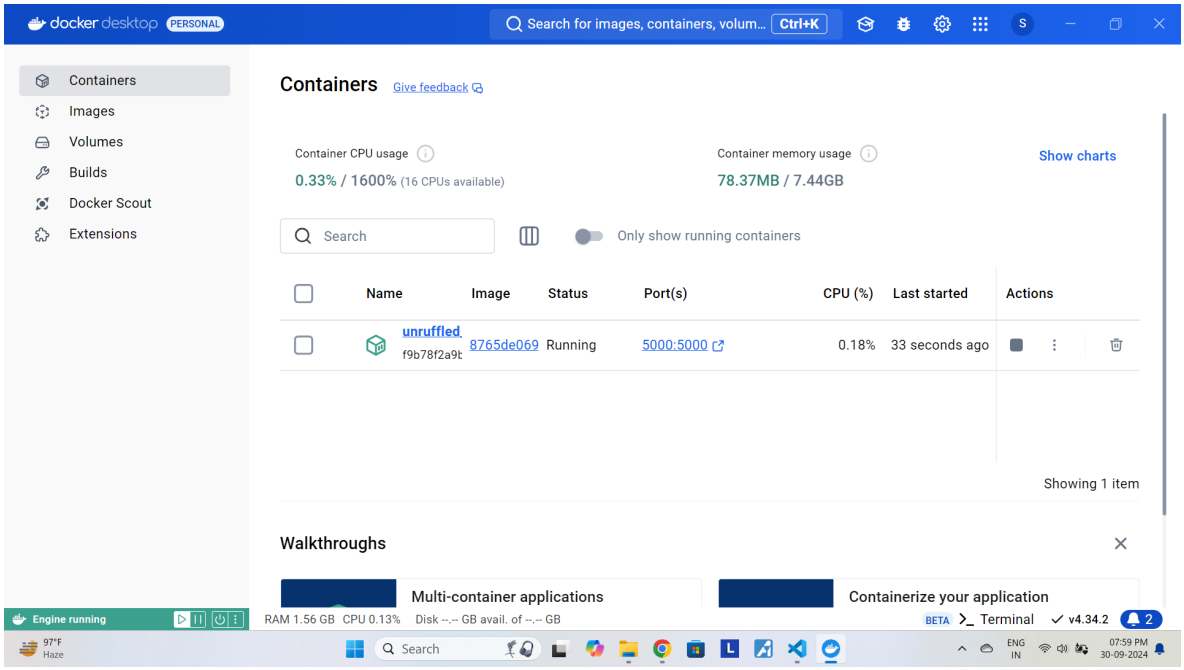
Core Endpoints:

1. **/ or /home**: Displays available rides according to the user's gender preference if authenticated, or presents login/signup options if not.
2. **/register and /login**: Facilitates user registration and authentication. These routes handle POST requests to submit user credentials and execute actions such as storing users in the database or validating credentials.
3. **/ride/new**: Facilitates the creation of rides. Upon posting a ride, a user is automatically designated as a passenger, resulting in a decrease in the number of available seats.
4. **/ride/join/¡int:ride_id¿**: Permits users to join a current ride if they satisfy the gender requirements and seats are available.
5. **/logout**: Terminates the user's session and purges any cached data to prevent unwanted access.
6. **/joined_rides**: Displays a list of rides the user has joined or created.
7. **/delete_ride/¡int:ride_id¿**: Allows users to delete a ride that they created.
8. **/opt_out/¡int:ride_id¿**: Enables users to leave a ride they have joined, freeing up a seat.
9. **/search_rides**: Allows users to find available rides and select the one they prefer.

7 Containerization:

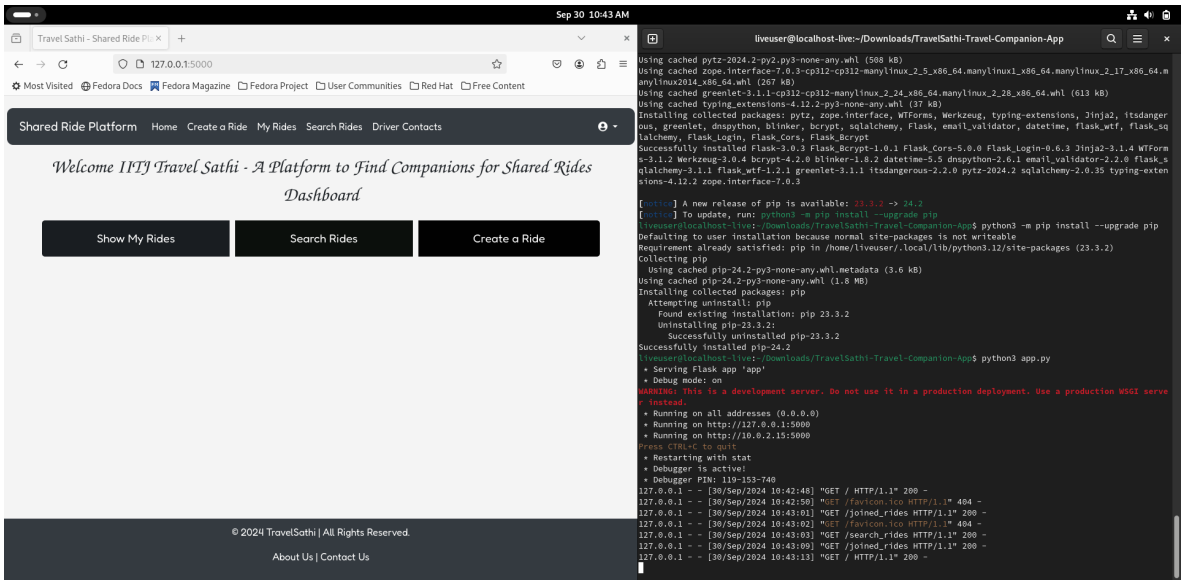
For containerizing our application,we use docker.

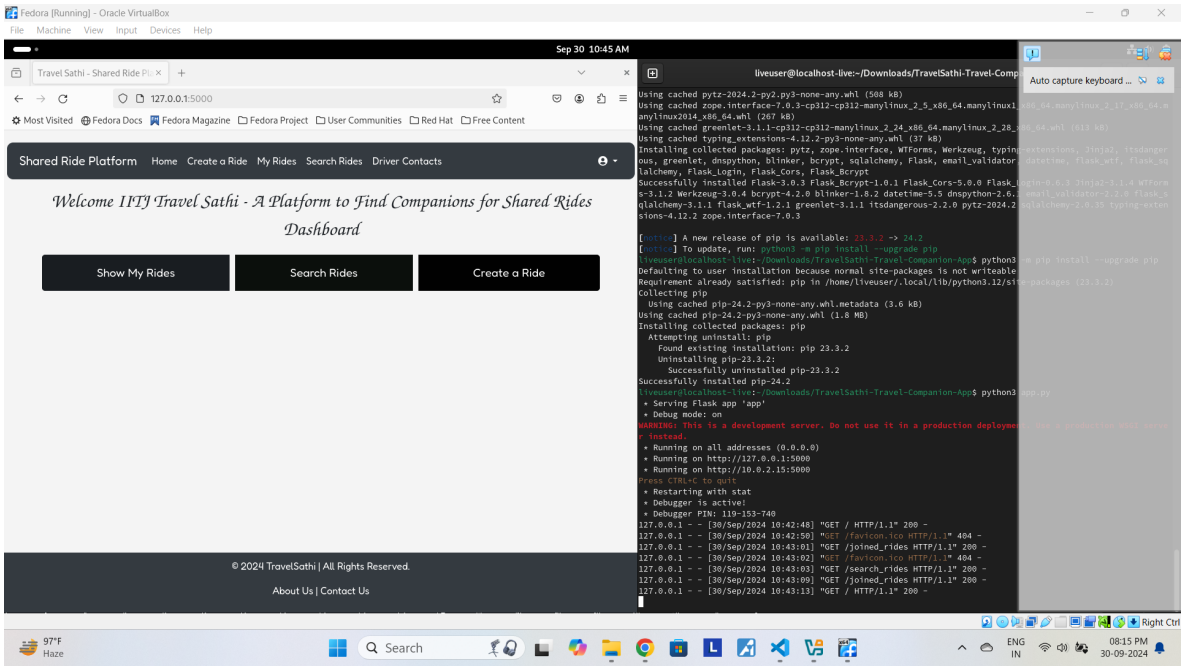



docker desktop PERSONAL
Search for images, containers, volum... Ctrl+K
Containers
Images
Volumes
Builds
Docker Scout
Extensions
Containers
Give feedback
Container CPU usage
0.33% / 1600% (16 CPUs available)
Container memory usage
78.37MB / 7.44GB
Show charts
Search
Only show running containers
Name
Image
Status
Port(s)
CPU (%)
Last started
Actions
unruffled
f9b78f2a9b
8765de069
Running
5000:5000
0.18%
33 seconds ago
Showing 1 item
Walkthroughs
Multi-container applications
Containerize your application
Engine running
RAM 1.56 GB CPU 0.13% Disk --.-- GB avail. of --.-- GB
BETA
Terminal
v4.34.2
2
97°F
Haze
Search
ENG
IN
07:59 PM
30-09-2024

8 Virtualization:

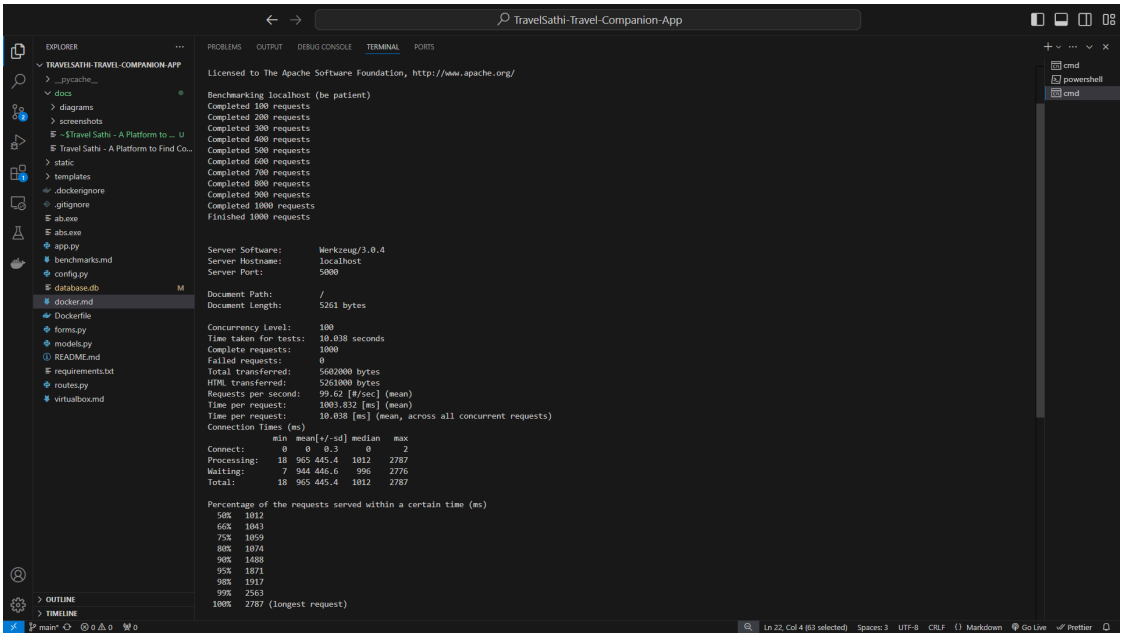
For virtualization,we first install virtualbox on which we run a Fedora based Virtual Machine.On that machine we deploy our website.

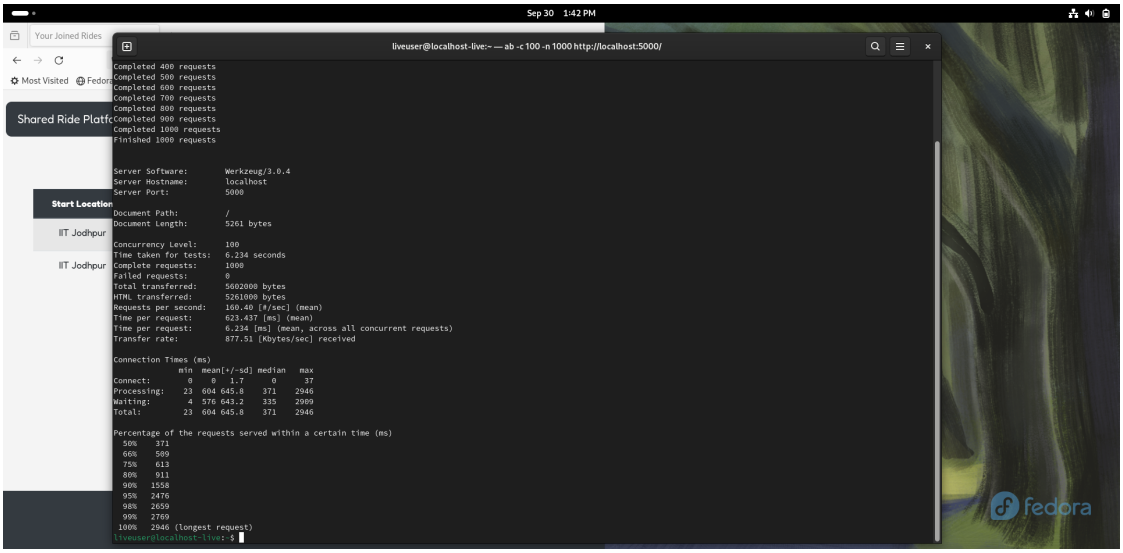




9 Testing Strategy and Results:

1. For Testing purposes we use apache benchmarks
2. Before running the test,we put constraints of 1 cpu and 2gb ram on both to ensure that the conditions are same.
3. Then we run the following code "ab -c 100 -n 1000 http://localhost:5000/" which means 100 concurrent connections and 1000 requests to the pages respectively.
4. We measure the performance of both based on this test and the results are as follows:-





10 Future Work:

1. We plan to expand it to nearby campuses like NIFT and Ayurvedic college.

2. Deployment on cloud platforms such as Google Cloud.

3. Notification alerts when someone joins or exits a ride using google firebase.

4. Moving to a better tech stack.

11 Conclusion:

As we can see,in our experiment containerization gave better metrics than virtualization.