

# 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**



॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

**Indian Institute of Technology Jodhpur**  
**Department of Computer Science and Engineering**

*September, 2024*

# Contents

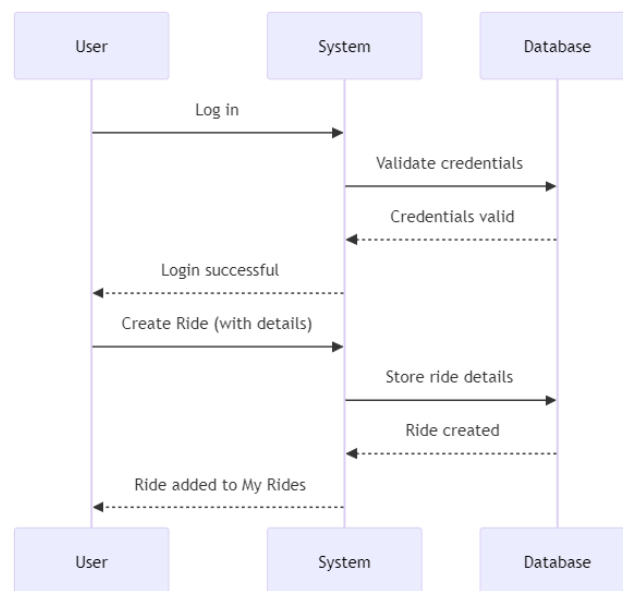
<b>1 Objective</b>	<b>1</b>
<b>2 System Specifications</b>	<b>1</b>
<b>3 Functional Requirements:</b>	<b>3</b>
<b>4 UI Design:</b>	<b>4</b>
4.1 Structure of the HTML files:- . . . . .	4
4.2 Screenshots of the pages:- . . . . .	5
<b>5 Database Design:</b>	<b>8</b>
<b>6 Analysis of Routes:</b>	<b>9</b>
<b>7 Containerization:</b>	<b>9</b>
<b>8 Virtualization:</b>	<b>11</b>
<b>9 Testing Strategy and Results:</b>	<b>12</b>
<b>10 Future Work:</b>	<b>12</b>
<b>11 Conclusion:</b>	<b>13</b>

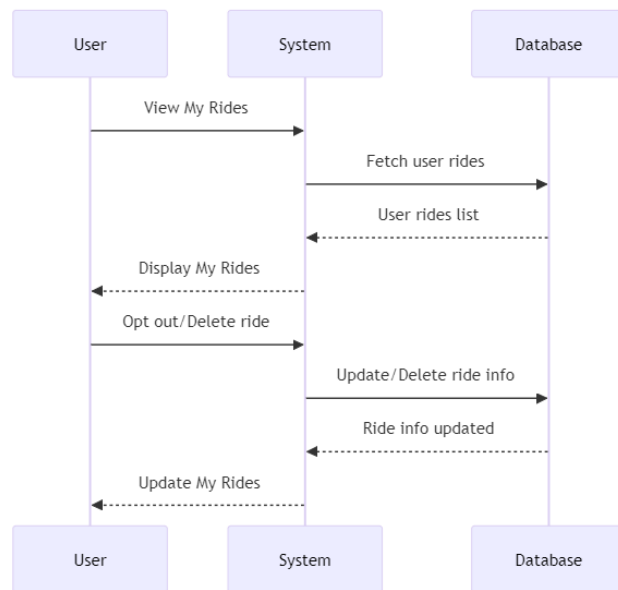
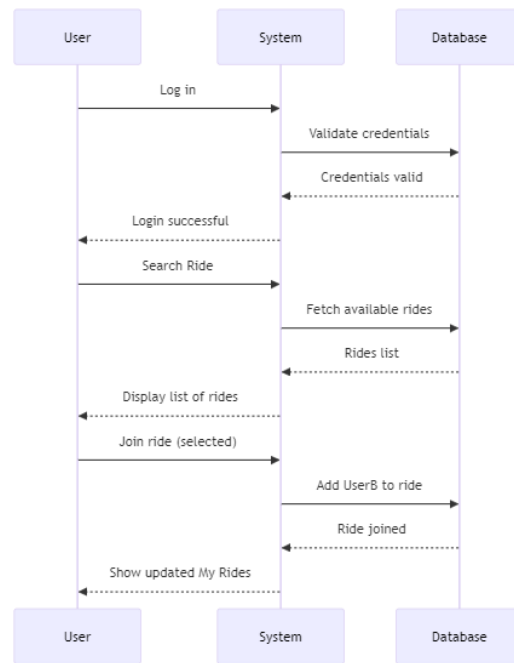
# 1 Objective

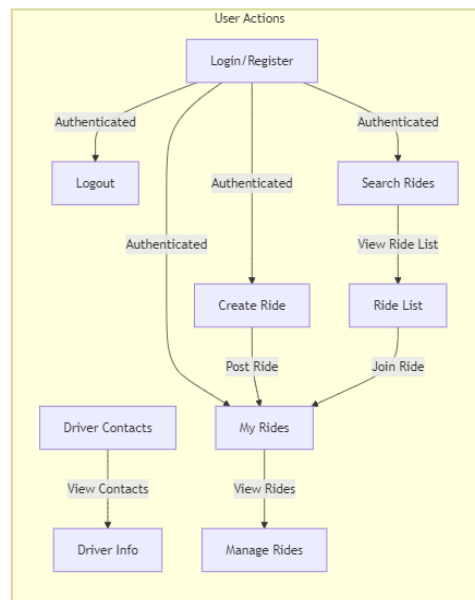
The project aims to create a secure online platform that facilitates shared transportation for college students, starting with rides between key locations like IIT Jodhpur and Jodhpur Junction. Users can post when they are looking for companions for a ride, and others can search for available rides to 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.







#### 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 PlatformHomeCreate a RideMy RidesSearch RidesDriver Contacts

Rickshaw Drivers

Below is a list of reliable rickshaw drivers you can contact for your rides:

Rajesh Kumar

Vehicle: Auto Rickshaw  
Contact: +91 9876543210  
Vehicle No: JDH00ER1  
Availability: 8 AM - 8 PM  
Languages: Hindi, English

Suraj Singh

Vehicle: E-Rickshaw  
Contact: +91 9876543211  
Vehicle No: JDH00ER2  
Availability: 9 AM - 9 PM  
Languages: Hindi, Marwari

Anil Sharma

Vehicle: Auto Rickshaw  
Contact: +91 9876543212  
Vehicle No: JDH00AR1  
Availability: 7 AM - 7 PM  
Languages: Hindi, English

Suresh Verma

Vehicle: Auto Rickshaw  
Contact: +91 9876543213  
Vehicle No: JDH00AR2  
Availability: 6 AM - 10 PM  
Languages: Hindi

© 2024 TravelSathi | All Rights Reserved.

About Us | Contact Us

Shared Ride PlatformHomeCreate a RideMy RidesSearch RidesDriver Contacts

Available Rides

Start Location	End Location	Departure Time	Seats Available	Cost per Head	Gender Preference	Action
Jodhpur Airport	IIT Jodhpur	01/10/2024 08:00 AM	1	350.0	Co-ed	<a href="#">Join Ride</a>

© 2024 TravelSathi | All Rights Reserved.

About Us | Contact Us

Shared Ride PlatformHomeCreate a RideMy RidesSearch RidesDriver Contacts

Your Joined Rides

Start Location	End Location	Departure Time	Available Seats	Action
IIT Jodhpur	Jodhpur Airport	30/09/2024 10:00 AM	2	<div><div>Opt Out</div><div>Delete Ride</div></div>

© 2024 TravelSathi | All Rights Reserved.

About Us | Contact Us

5

Shared Ride Platform

Home

Create a Ride

My Rides

Search Rides

Driver Contacts

9

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

9

Create a New Ride

Sumit@iitj.ac.in

Logout

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



## 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](#)

## 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:

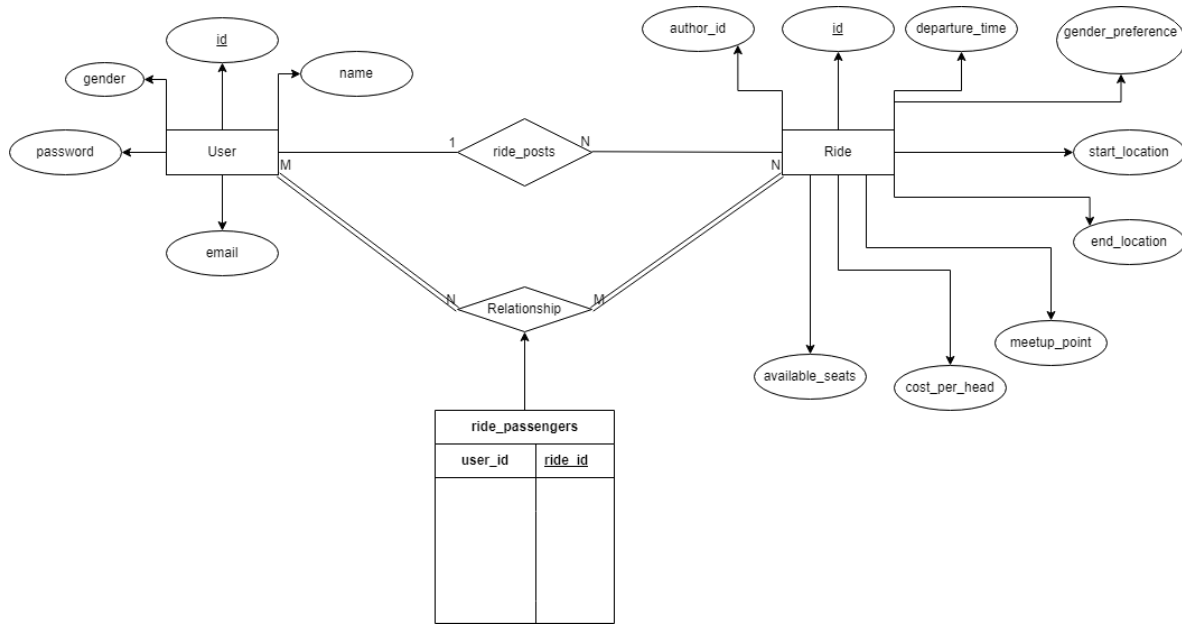


Table : User

### 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

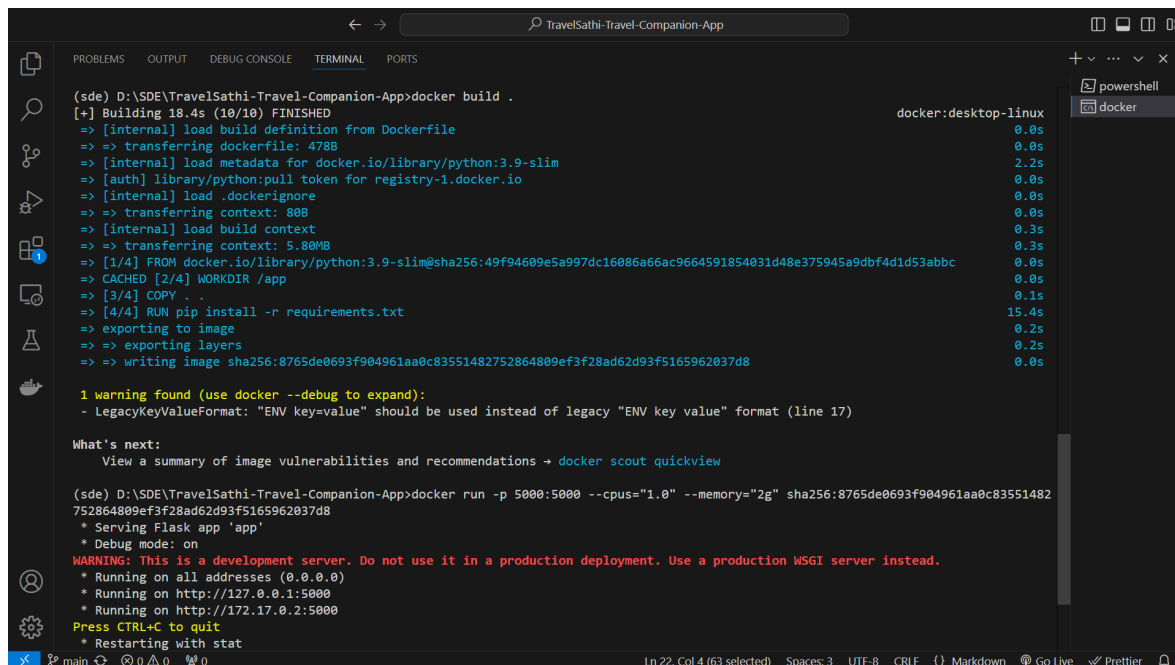
## 6 Analysis of Routes:

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.

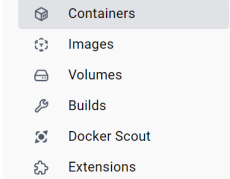


```
(sde) D:\SDE\TravelSathi-Travel-Companion-App>docker build .
[+] Building 18.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 478B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 80B
=> [internal] load build context
=> => transferring context: 5.80MB
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN pip install -r requirements.txt
=> exporting to image
=> exporting layers
=> writing image sha256:8765de0693f904961aa0c83551482752864809ef3f28ad62d93f5165962037d8

1 warning found (use docker --debug to expand):
- LegacyKeyValueFormat: "ENV key=value" should be used instead of legacy "ENV key value" format (line 17)

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview

(sde) D:\SDE\TravelSathi-Travel-Companion-App>docker run -p 5000:5000 --cpus="1.0" --memory="2g" sha256:8765de0693f904961aa0c83551482752864809ef3f28ad62d93f5165962037d8
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
```



## Containers [Give feedback](#)

Container CPU usage ⓘ

0.33% / 1600% (16 CPUs available)









Container memory usage 

78.37MB / 7.44GB

[Show charts](#)

Q Search

☒ Only show running containers

	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
	 <a href="#">unruffled</a>	 <a href="#">8765de069</a>	Running	<a href="#">5000:5000</a> 	0.18%	33 seconds ago	  

Showing 1 item

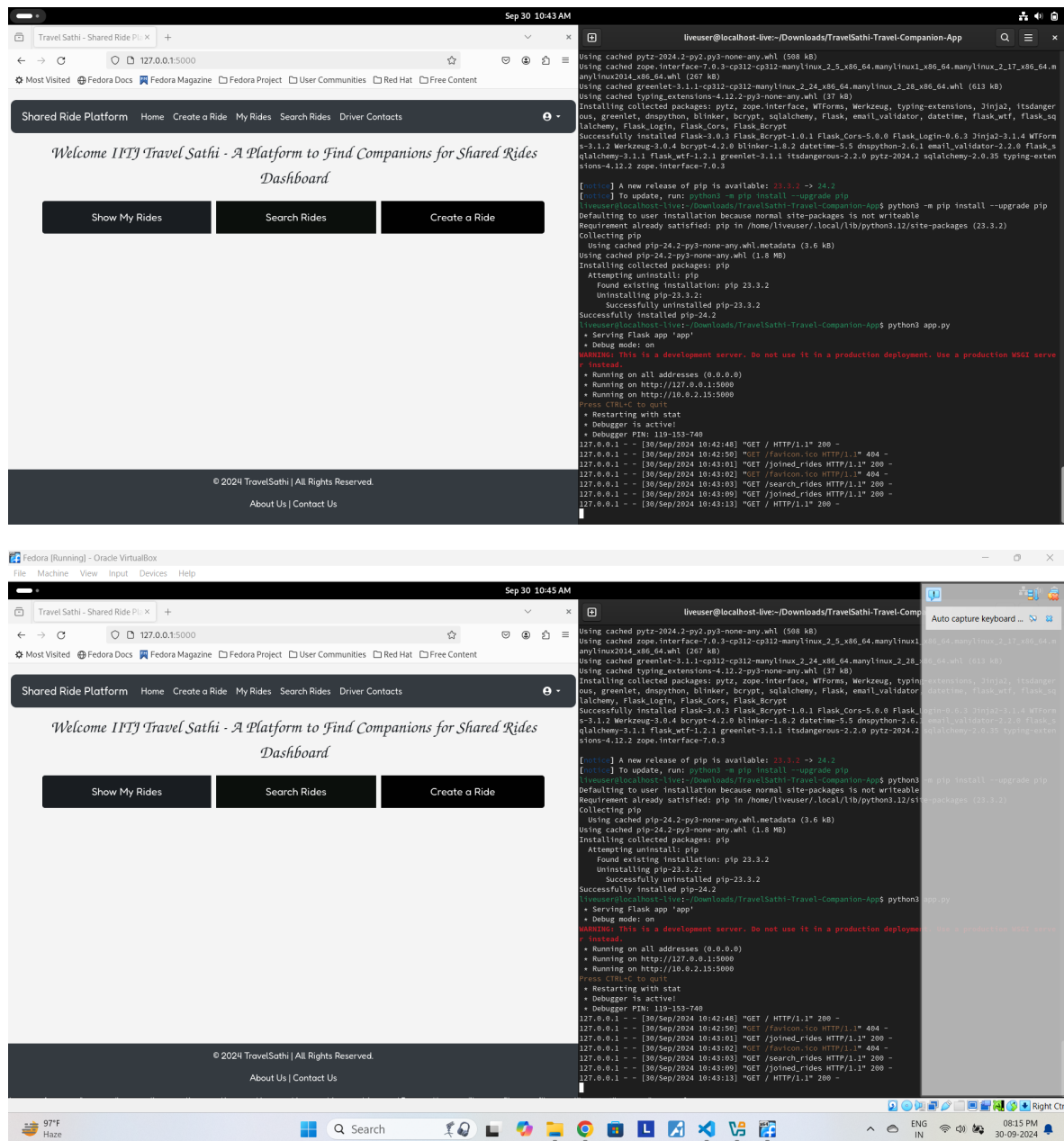
## Walkthroughs


Multi-container applications


Containerize your application

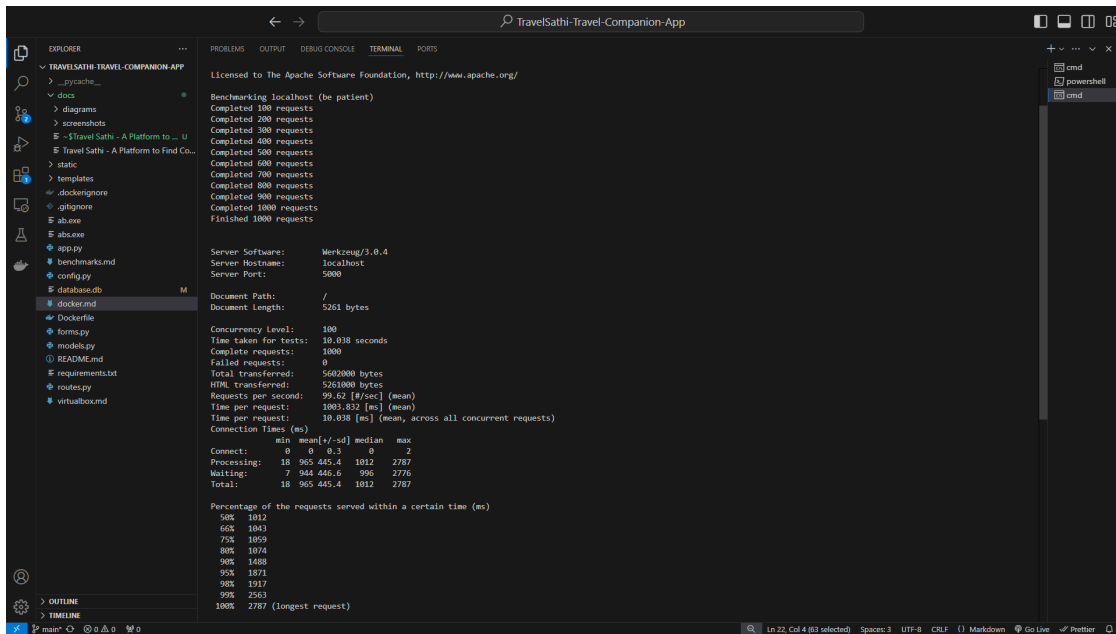
## 8 Virtualization:

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:-



```
TravelSathi-Travel-Companion-App
Licensed to The Apache Software Foundation, http://www.apache.org/

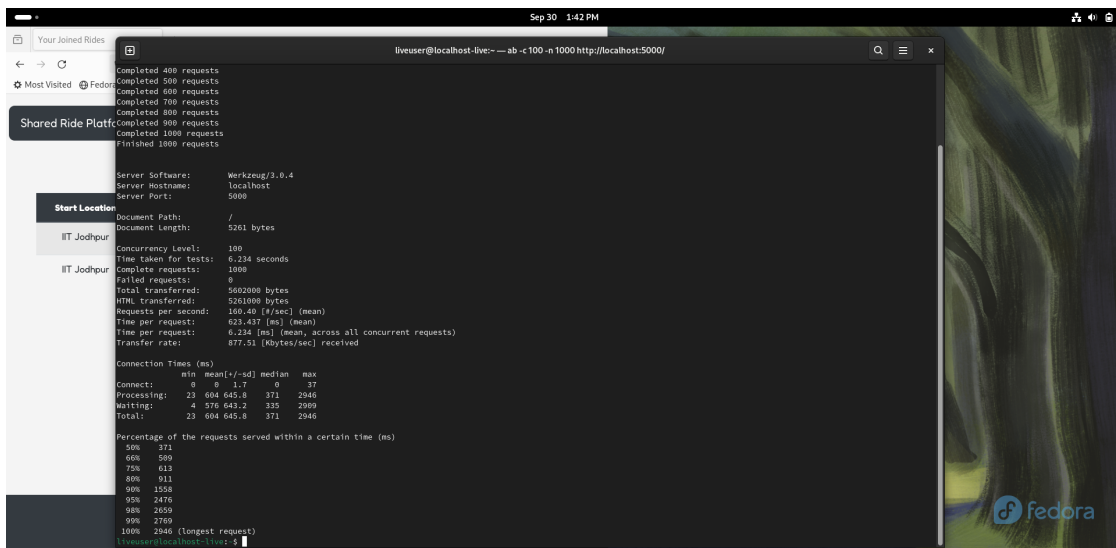
Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software: Werkzeug/3.0.4
Server Hostname: localhost
Server Port: 5000

Document Path: /
Document Length: 5261 bytes

Concurrency Level: 100
Time taken for tests: 10.038 seconds
Complete requests: 1000
Failed requests: 0
Total transferred: 5262000 bytes
HTML transferred: 5261000 bytes
Requests per second: 99.62 [#/sec] (mean)
Time per request: 1003.832 [ms] (mean)
Time per request: 10.038 [ms] (mean, across all concurrent requests)
Connection Times (ms)
  min mean[+/-sd] median max
Connect: 0 0 0.3 0 2
Processing: 18 965 445.4 1012 2787
Waiting: 7 944 446.6 996 2776
Total: 18 965 445.4 1012 2787

Percentage of the requests served within a certain time (ms)
 50% 1012
 66% 1043
 75% 1059
 80% 1074
 90% 1488
 95% 1871
 98% 1917
 99% 2543
100% 2787 (longest request)
```



```
liveuser@localhost-livc:~$ ab -c 100 -n 1000 http://localhost:5000/

Completed 400 requests
Completed 600 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software: Werkzeug/3.0.4
Server Hostname: localhost
Server Port: 5000

Document Path: /
Document Length: 5261 bytes

Concurrency Level: 100
Time taken for tests: 6.234 seconds
Complete requests: 1000
Failed requests: 0
Total transferred: 5262000 bytes
HTML transferred: 5261000 bytes
Requests per second: 160.40 [#/sec] (mean)
Time per request: 623.437 [ms] (mean)
Time per request: 6.234 [ms] (mean, across all concurrent requests)
Transfer rate: 877.51 [Kbytes/sec] received

Connection Times (ms)
  min mean[+/-sd] median max
Connect: 0 0 1.7 0 37
Processing: 23 604 645.8 371 2946
Waiting: 4 576 643.2 335 2909
Total: 23 604 645.8 371 2946

Percentage of the requests served within a certain time (ms)
 50% 371
 66% 589
 75% 613
 80% 911
 90% 1538
 95% 2476
 98% 2659
 99% 2769
100% 2946 (longest request)
```

## 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.