Team Project
Deliverable 5 – Project Phase 3
CSCE 5430 (Spring 2024)

Implementation Phase 3: Optimization

Requirements:

a) Search Engine

After the successful development of the code for our Dreamscape destinations website, we headed to perform SEO Optimization going through all the top notched SEO policies and roles. So, we looked through several articles, which are listed below. By employing these techniques, we increased website visibility in any browser, allowing both customers and vendors to benefit from it.

https://ahrefs.com/blog/ecommerce-seo/

i. Front End Optimization:

In the previous phase, phase 2, we held regular peer meetings and solicited feedback from various users to improve our website. While these enhancements were not originally planned for phase 3, we decided to implement them as this is the semester's final phase. We used dynamic animations to enhance the user experience.

ii. OS Adaptability:

We've updated our website to dynamically adjust to the user's operating system preferences. For example, our website now provides optimal viewing and interaction experiences across a variety of devices. Previously only available on desktops, users can now access and interact with our website on mobile devices without experiencing any graphical issues. This ensures that users can access our website from any device with ease.

b) Backend:

i. Blogs/Feedback:

The backend development for DreamEscape's blog feature was focused on laying a solid foundation for seamless integration with the front end. This included creating a strong GraphQL schema to facilitate data manipulation and integrating MongoDB as a scalable database solution. To secure API endpoints, authentication and authorization mechanisms based on JWT were implemented, while data integrity and security were maintained through extensive validation and sanitization techniques. Query performance was improved by using indexing strategies and query optimization techniques to improve scalability and responsiveness, especially during peak usage. Furthermore, robust error handling and logging mechanisms were implemented to enable efficient troubleshooting and debugging, ensuring the backend infrastructure's dependability and stability.

ii. Authentication:

The AuthService class in DreamEscape's blogging platform handles user authentication seamlessly. It includes methods such as getProfile(), which decodes user data from authentication tokens, loggedIn(), which determines whether a user is logged in, and isTokenExpired(), which checks token expiration. getToken() returns authentication tokens, whereas login(idToken) stores tokens for authenticated sessions and redirects users. logout() securely logs users out by clearing tokens and profile data before reloading the page. This centralized approach ensures streamlined and secure authentication, thereby improving maintainability, security, and user experience.

iii. Contact us:

The "Contact Us" page on DreamEscape Destination website offers visitors a convenient way to reach out for inquiries, feedback, or collaboration opportunities. It provides various contact options, including a contact form, email address, and phone number, ensuring accessibility for all visitors. The page begins with a welcoming message, outlining the purpose of the contact page and the types of inquiries welcomed. Clear and concise instructions guide visitors on how to get in touch effectively. It’s a communication bridge between the vendors and the customers to know more about the packages and destnations , their budgets and availiability. Overall, the "Contact Us" page is designed to facilitate seamless communication between Dreamscape Destination and its visitors, enhancing user experience and fostering engagement.

iv. FAQ:

In our SEO optimization efforts, we've incorporated a Frequently Asked Questions (FAQ) page to provide clarity on our website's operations. These questions address common concerns users typically encounter when visiting a travel destination or airflight booking website , such as delivery, returns, and payment processes.

v. TOC:

To ensure transparency and provide guidance for both customers and vendors, we've introduced a Terms of Conditions page. Through extensive discussions and deliberations, we've outlined the core perspectives and incorporated them into this support page. This not only serves as a reference point for users but also aids in resolving disputes according to the agreed terms.

vi. Privacy Policy:

As part of our commitment to safeguarding user data, including personal and payment information, we prioritize data privacy within our online community. Users can confidently provide their bank details only if they are assured of the security of payment gateway or any dispute between the travel agents and the customers as we are a third party perosns just hosting the website connecting them both. To underscore our dedication to privacy and security, we've implemented stringent security measures and established a Privacy Policy page to reassure users that their data is handled responsibly and protected from misuse.

C) Database:

MongoDB Database with Mongoose: Flexible and Scalable Data Storage. Our project takes advantage of MongoDB, a NoSQL database, for persistent data storage via the Mongoose library. MongoDB's schema-less design provides unparalleled flexibility, making it an excellent choice for storing diverse and changing data types such as user profiles, blog content, travel destinations, and booking information.

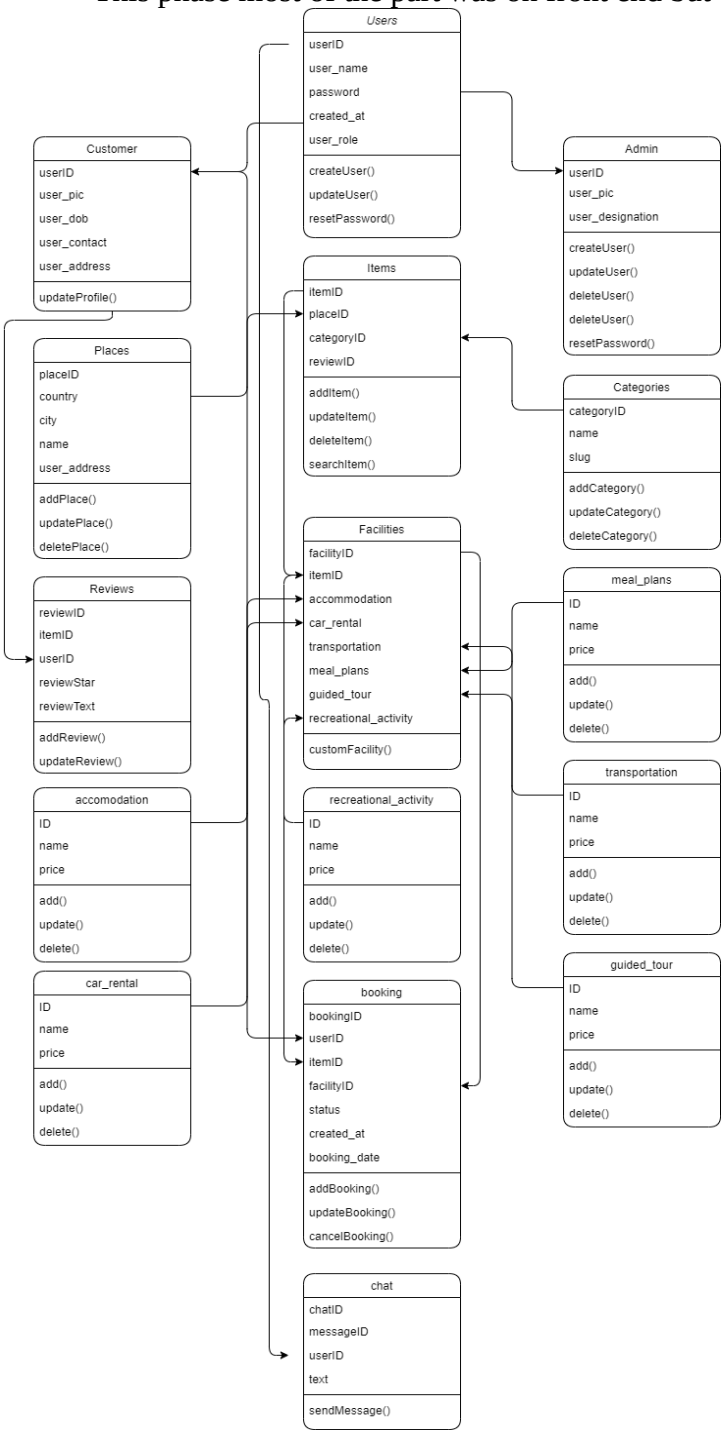
MongoDB was chosen because of its ability to seamlessly accommodate changing data requirements. With no strict schema constraints, rapid development and iteration are effortless. Furthermore, MongoDB excels at handling hierarchical data structures via nested documents and arrays, which perfectly suits our project's data hierarchy requirements.

Mongoose, a MongoDB object modeling tool, improves our Node.js development workflow by incorporating schema validation, data mapping, and powerful utilities.

UML Diagrams:

1. Class Diagram:

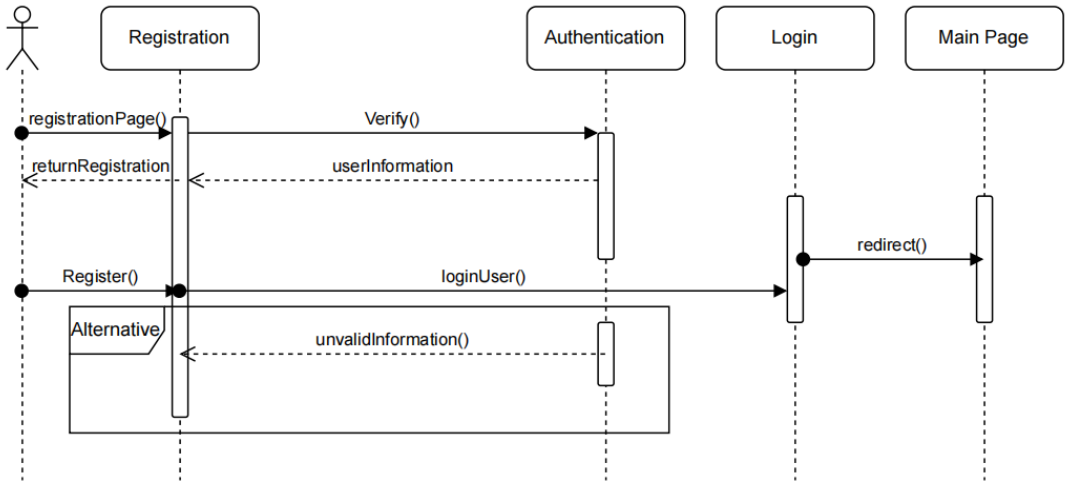
This phase most of the part was on front end but we have created whole UML:



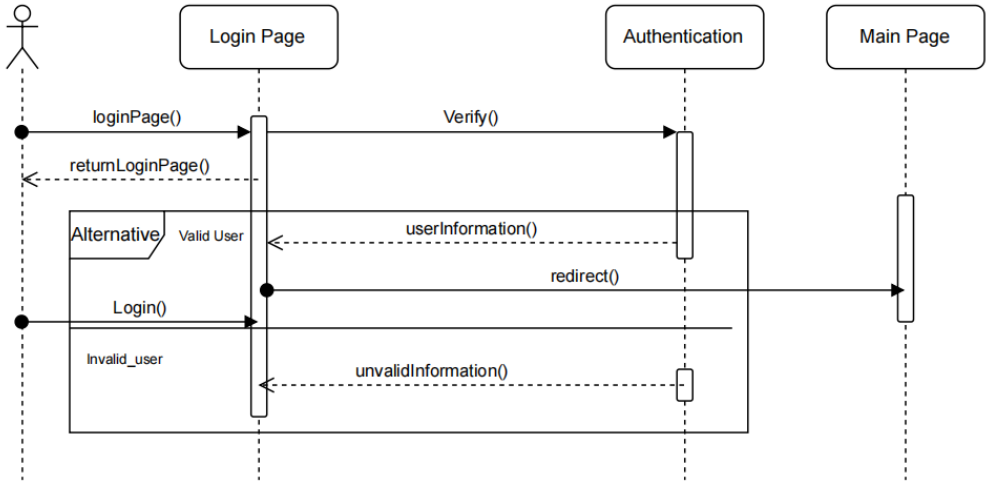
2. Sequence Diagram

Some Sequence diagrams of these functions are updated as the back-end development in this phase.

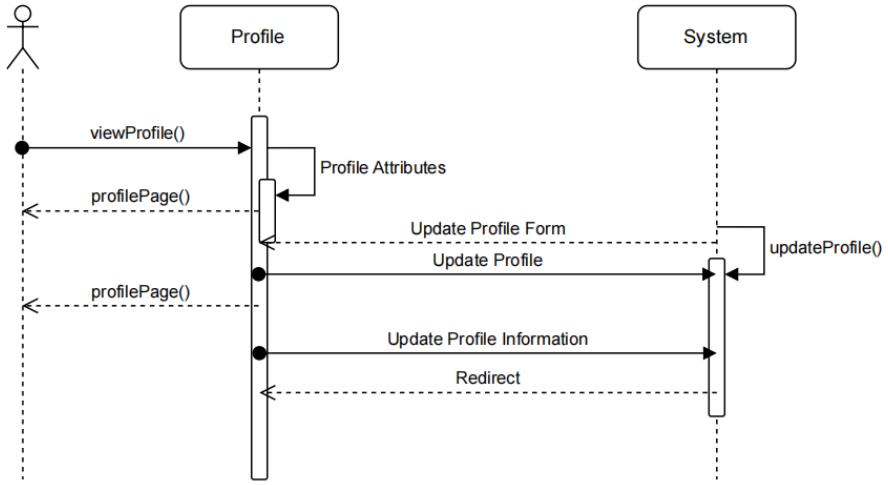
- Sign Up



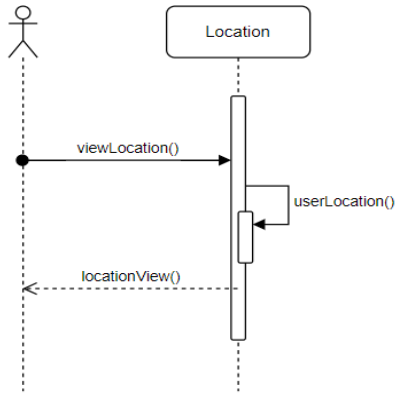
- Log in



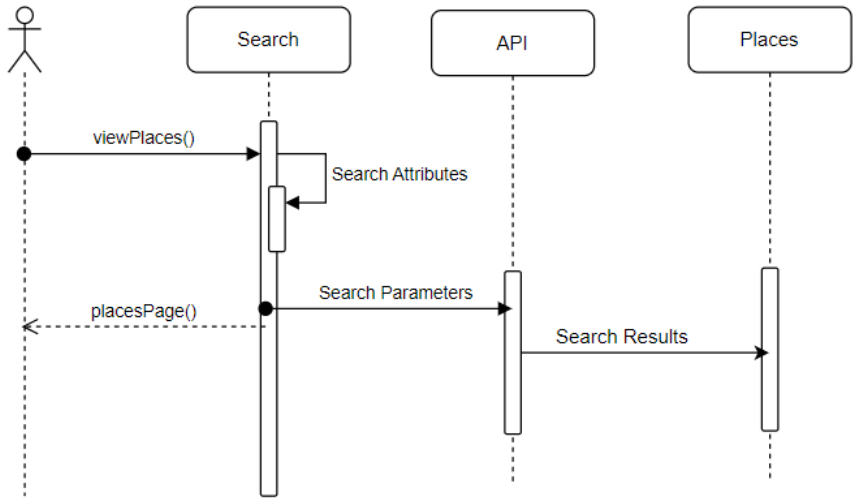
- Update Profile



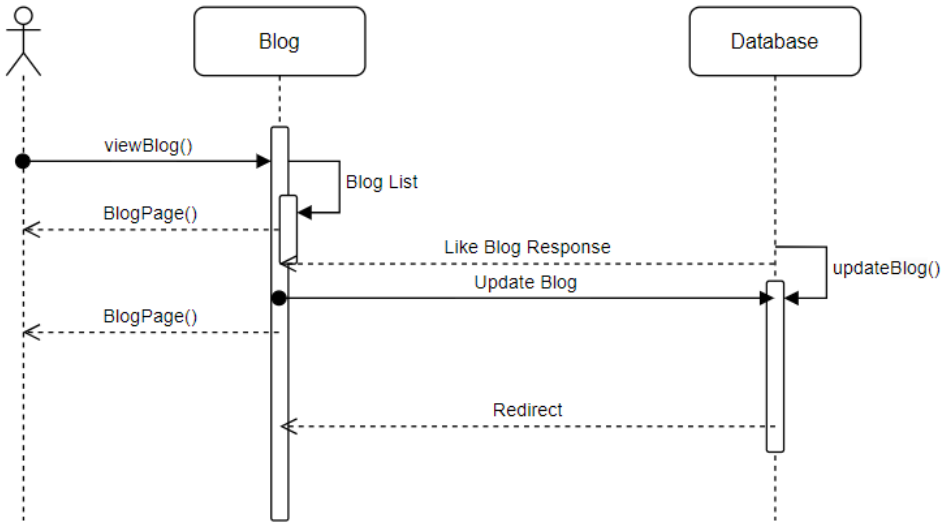
- User Location



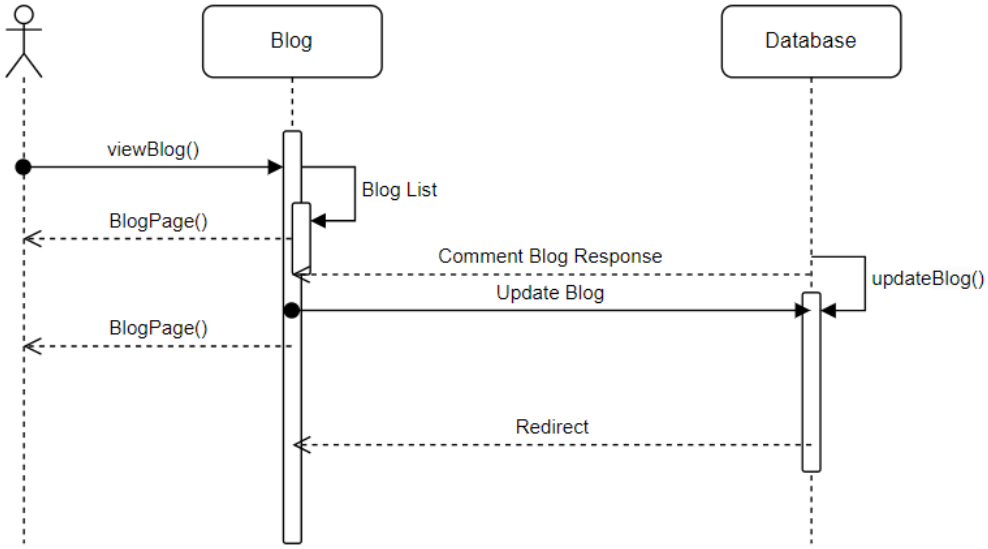
- Search Nearby Place



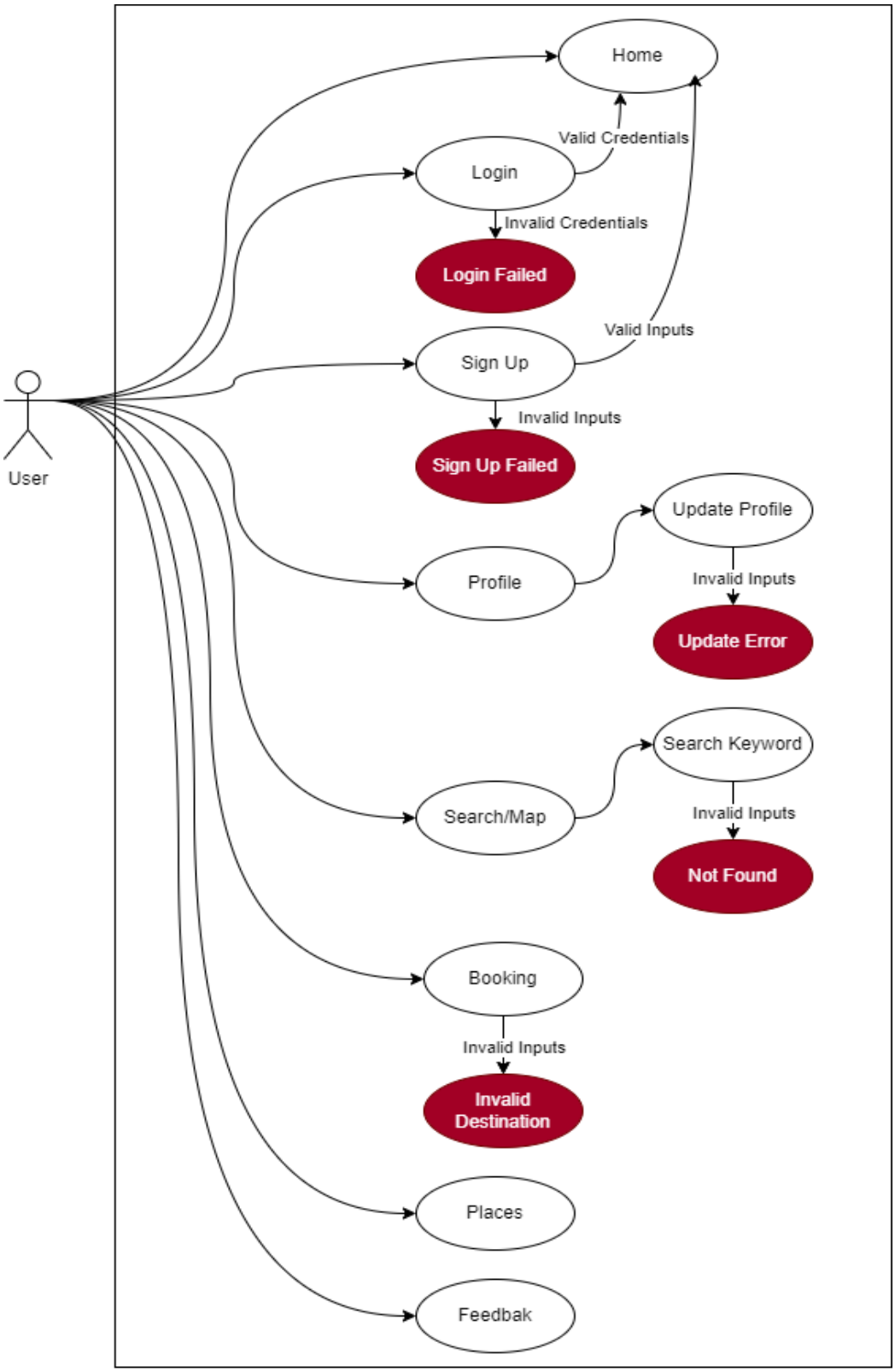
- Blog Like



- Blog Comment



3. Use Case Diagram



Test Cases:

a) Unit Testing

Functionality	Input	Output	Expected
Sign Up	Non-Unique Credentials	Error	Sign Up Error
Sign Up	Unique & Valid Credentials	Successfully Sign Up	Successfully Sign Up
Login	Wrong Credentials	Error	Credentials Error
Login	Authorized Credentials	Redirect to Home page	Successfully login
Book Tickets	Try adding destinations and date fields then then checking their flight availability	Error	Will show the exact place, where input is missing, and it need to be filed.
Book destinations	Fill all fields	Redirect to related description and price page	Will show price andlocation details
Update Profile	Fill in the Fields	Empty/Null	Then that space appears blank
Blog	Write about your experience going to a destination	The experience appears on your profile as a card visible to everyone in the blog	Visible to all
Conversation and Like	Like and comment on blog	Colour of the heart changes to red and count of the comment increases	
Conversation	Comment on some ones post and like them	Like doesn't get updated or comment doesn't apply	Like and comment are visible to all
Contact Us	User input values	Error from google	error

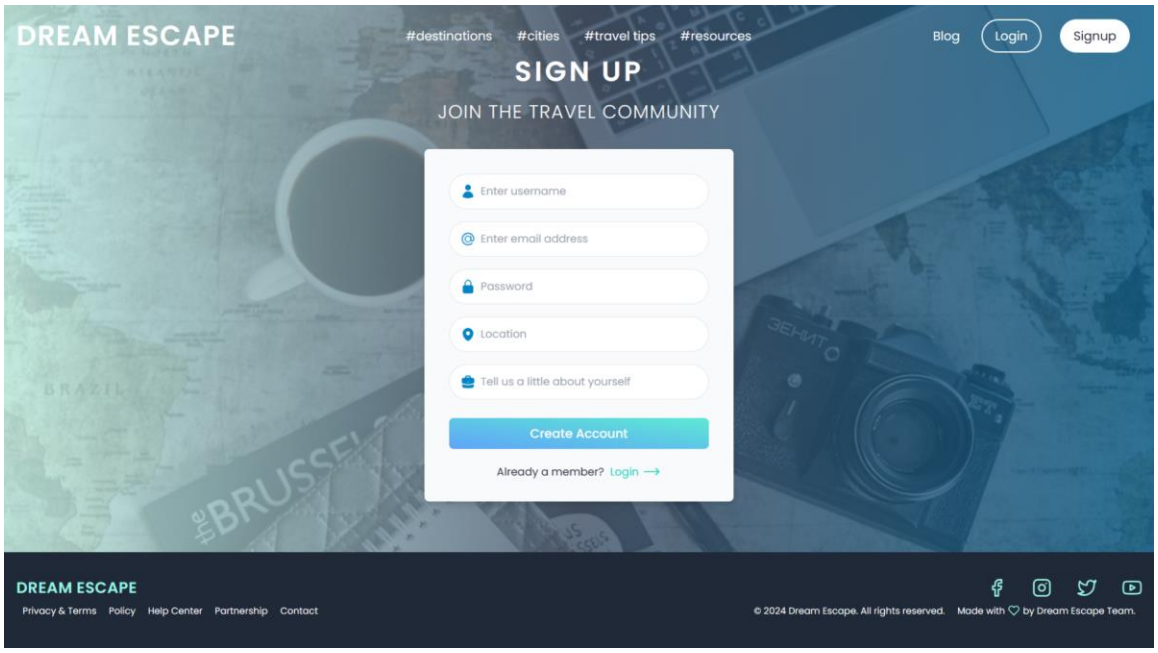
b) System Testing:

Method	Input	Affected Method	Expected Output	Output
Signup	SignUp Details entered correctly	profile_details	Profile added to database and user is logged in	Profile added to database and user is logged in
SignUp	SignUp Details entered not correctly		Profile not added to database. Error Appear.	Profile not added to database. Error Appear.
Login	Right Credentials	user_login, django_auth	User Logged in	User Logged in
Login	Wrong Credentials	user_login, django_auth	User not Logged in	User not Logged in
Add Booking	Booking Details	Booking List	All the list of flights between destinations	
initiate_conversation	Click on chat buttons	chat_view,	Created chat in database	Created chat in database
Review Blog	Submission of Review	Review_details	Added review to the database, and show on review page	Added review to the database, and show the blog page

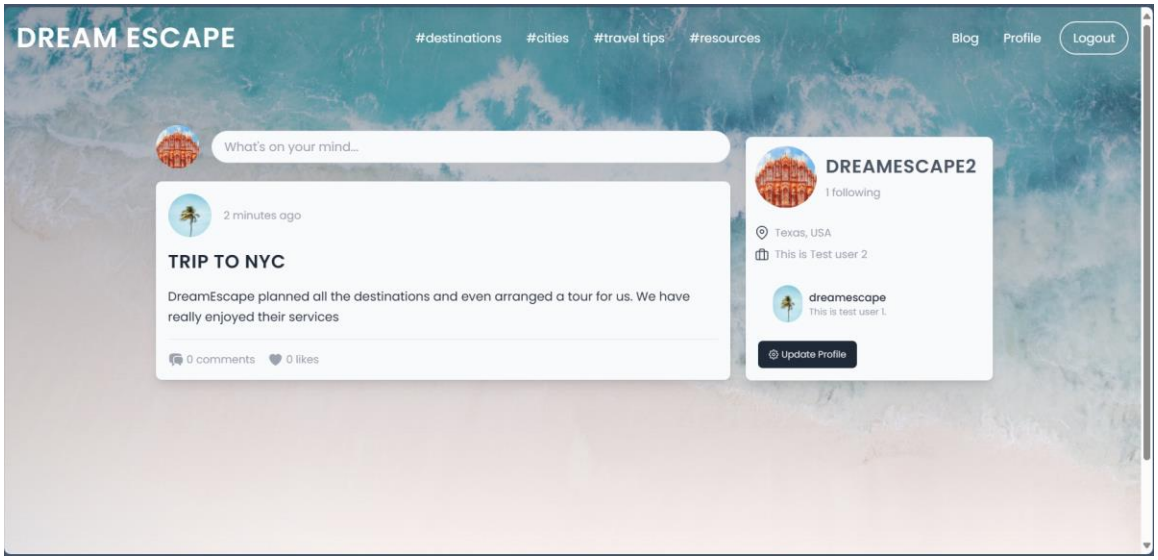
Front end Testing:

This phase mostly depends on the back end along with the new features that we added.

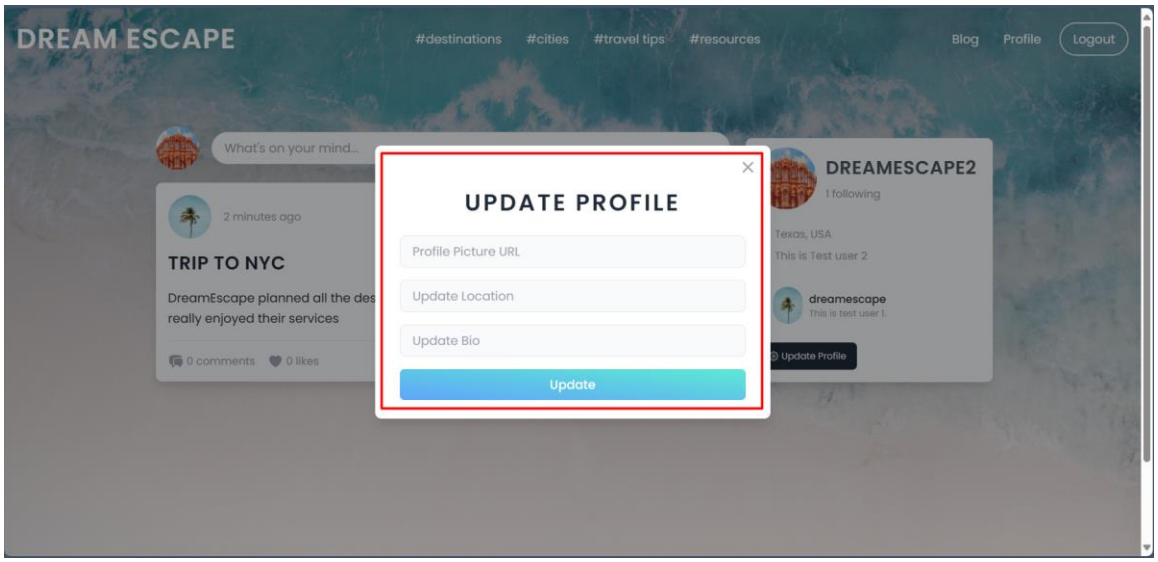
1. Signup Page



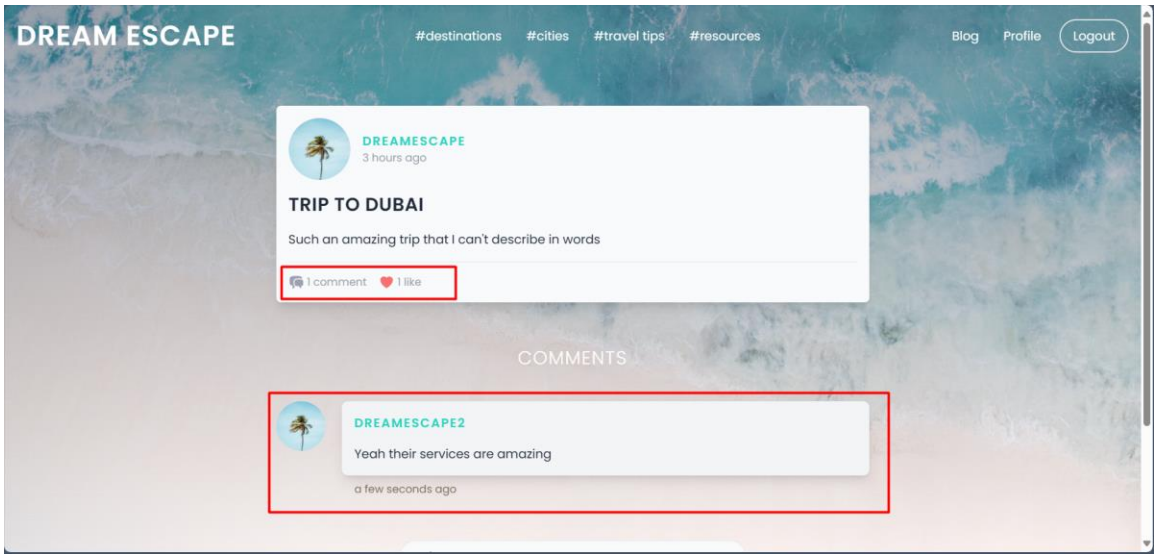
2. User Profile



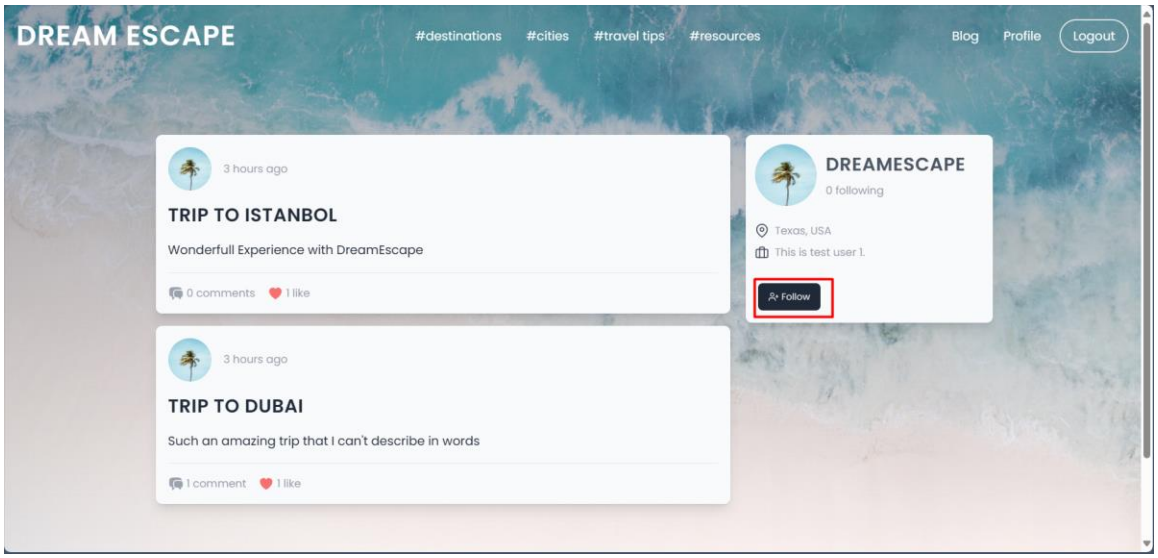
3. Profile Update:



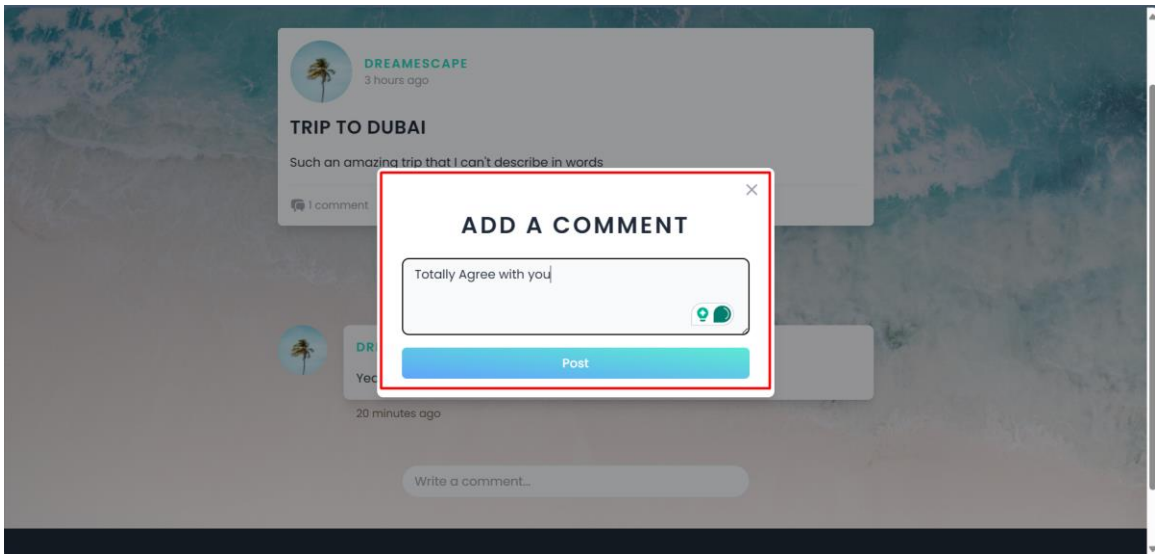
4. Like and comment a Blog:



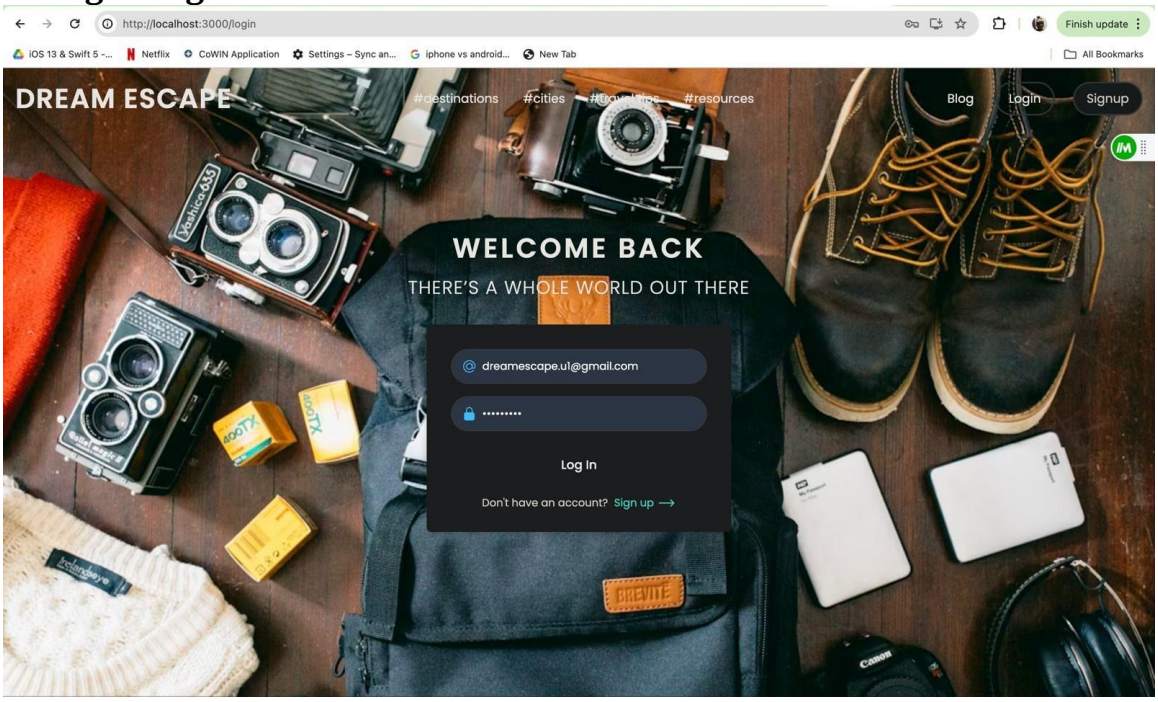
5. Follow a User.



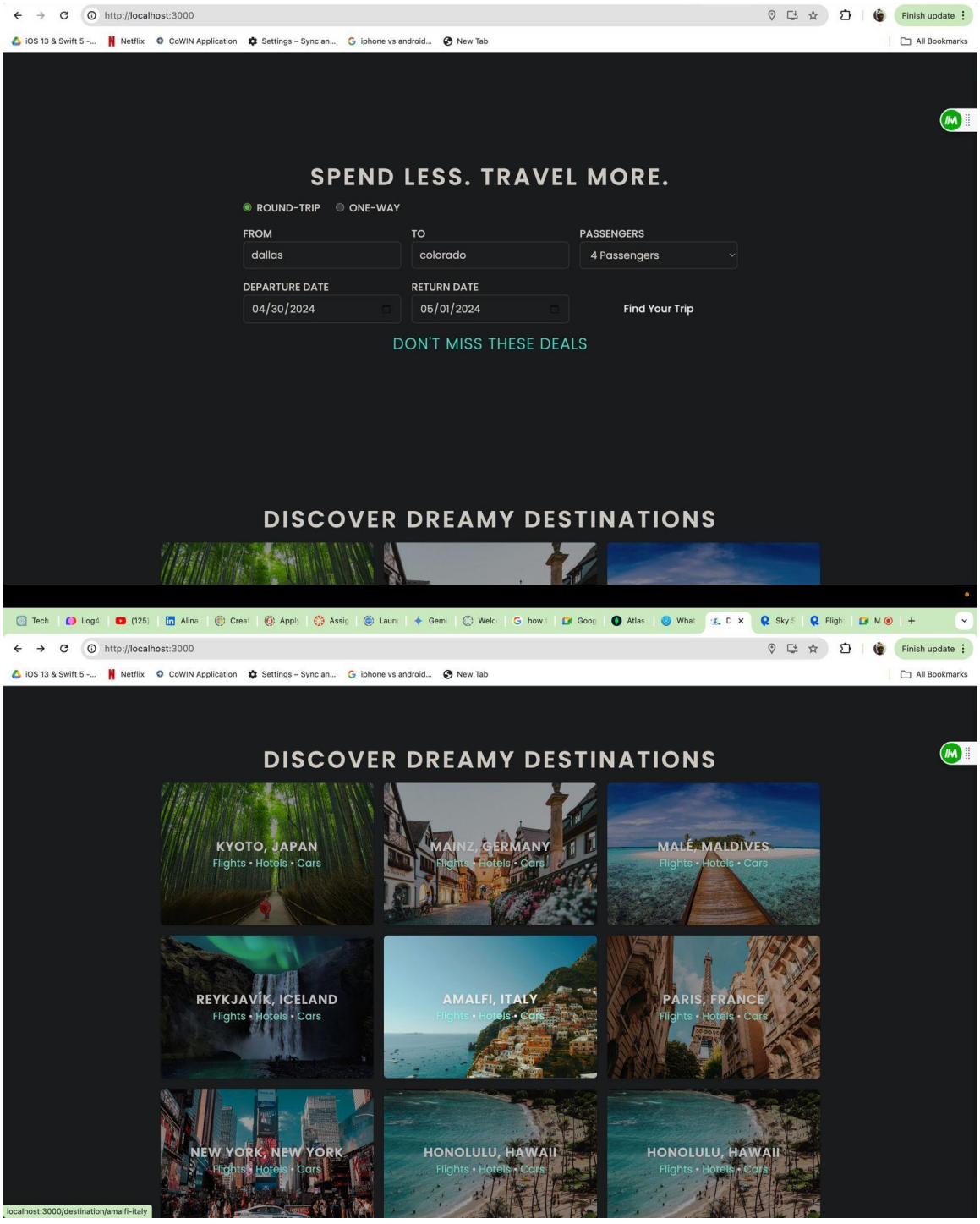
6. Add like/comment:



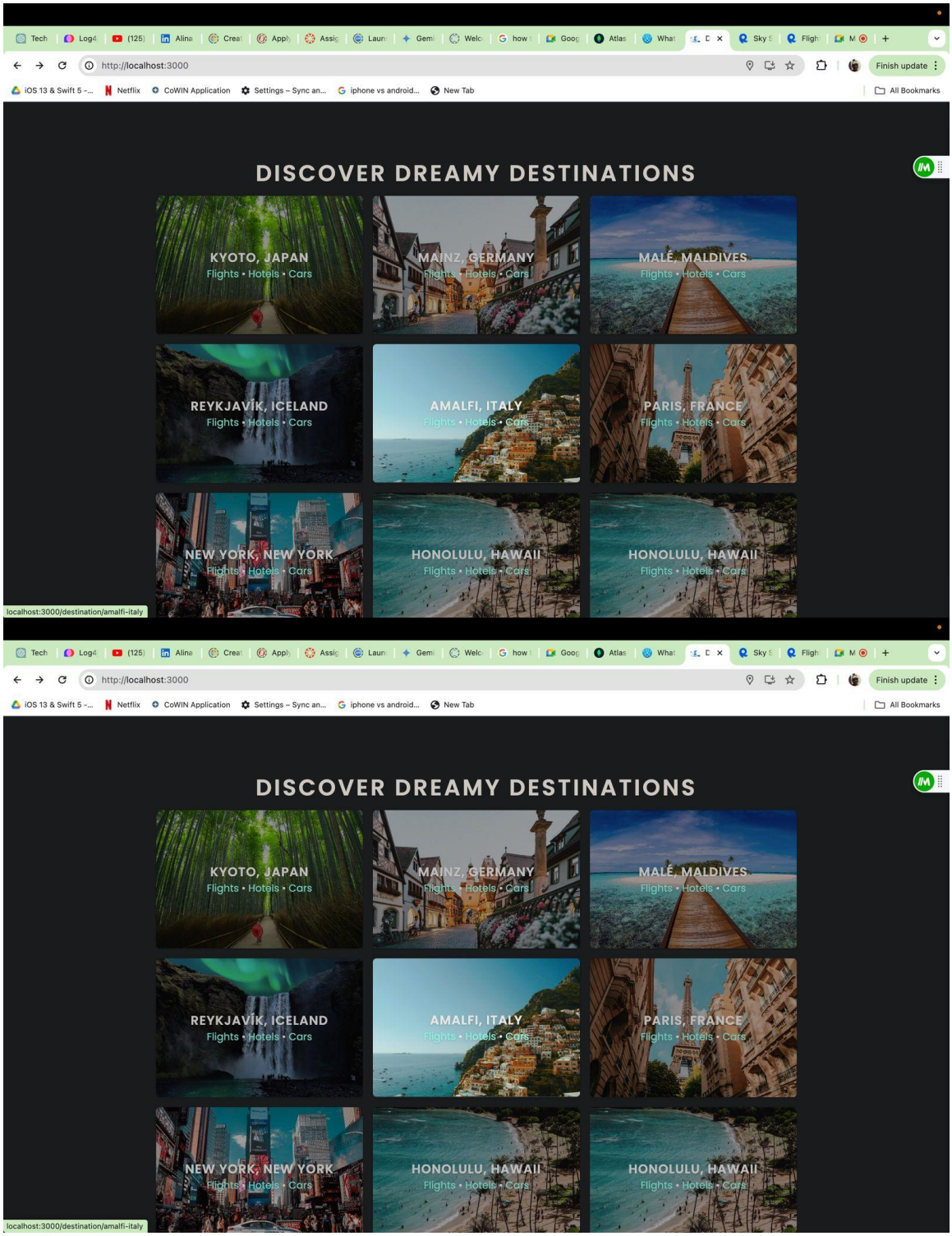
7. Login Page:



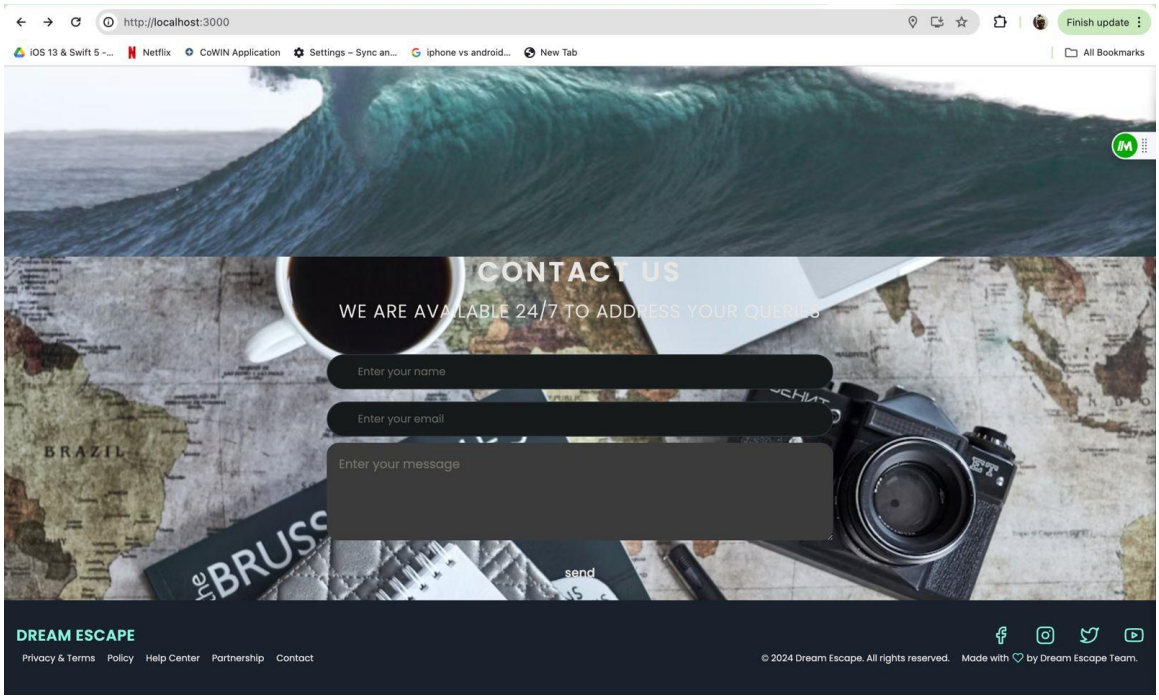
8. Booking:


http://localhost:3000
Finish update
iOS 13 & Swift 5 -...
Netflix
CoWIN Application
Settings – Sync an...
iphone vs android...
New Tab
All Bookmarks
SPEND LESS. TRAVEL MORE.
ROUND-TRIP
ONE-WAY
FROM
dallas
TO
colorado
PASSENGERS
4 Passengers
DEPARTURE DATE
04/30/2024
RETURN DATE
05/01/2024
Find Your Trip
DON'T MISS THESE DEALS
DISCOVER DREAMY DESTINATIONS
DISCOVER DREAMY DESTINATIONS
KYOTO, JAPAN
Flights • Hotels • Cars
MAINZ, GERMANY
Flights • Hotels • Cars
MALÉ, MALDIVES
Flights • Hotels • Cars
REYKJAVÍK, ICELAND
Flights • Hotels • Cars
AMALFI, ITALY
Flights • Hotels • Cars
PARIS, FRANCE
Flights • Hotels • Cars
NEW YORK, NEW YORK
Flights • Hotels • Cars
HONOLULU, HAWAII
Flights • Hotels • Cars
HONOLULU, HAWAII
Flights • Hotels • Cars
localhost:3000/destination/amalfi-italy

9. Recommendation:



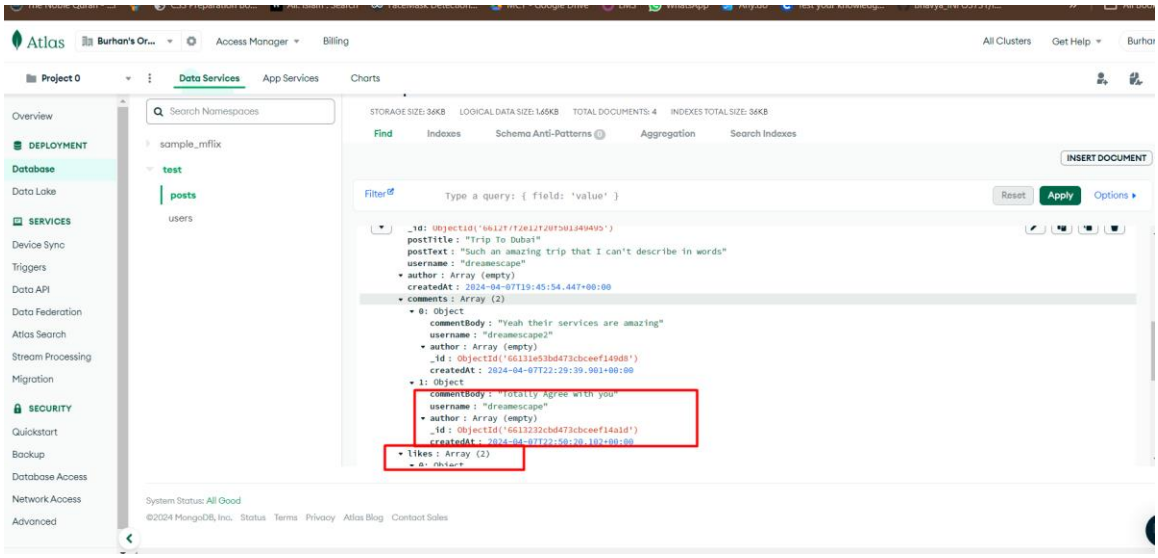
10. Contact Us:



Backend testing:

i. API Integration:

ii. Add/Like:



Installation User Manual:

- Prerequisites

  User must need to install Nodes.JS from the official website https://nodejs.org/en/download. Install as per the user operating system either it is Windows, Mac, or Linux. After downloading and installing the nodes, please make sure the NPM which is actually nodes package manager, is also installed correctly and to check this please run this command

  npm list -g

- Installation

  Once nodes is configured, you must need to install some modules that are being used in this project. To do this, open the client folder in the CMD and run following command

  npm install

  Once all modules are installed successfully now it's time to open the server folder inside CMD and repeat the process again. By these all-required modules for the front end and back end will be installed.

Program Run User Manual:

To run the program, user need to open both client and server folders in two separate CMD. After that in both cmd please write this command

npm start

This will start two ports. On port 3001 the server is running which is actually back end of our project and on port 3000 the front end will start running and you can access this by this url in your local browser

http://localhost:3000/

Accomplishments and Improvements:

In this project we set out to accomplish a destination website which included travel guide travel package and the booking of tickets for destinations. We have also made famous destinations where our users can investigate to travel and first of its kind to implement a blog among the community of our users to let them know the experience of travelling to famous destinations so that other users can know whether to choose to go to that place or not.
As a part of this we implemented sign up and login for the suers using our website so that the user has a profile page and can post about their experiences which every other user or customer on our website can choose based on their review and rating.

In the future we want to integrate payments to this and use machine learning models to give personalized suggestions and a chatbot to solve the issues and FAQ; s of our customers, which could enhance the customer experience.

Collaboration

Member name	Contribution description	Overall Contribution (%)
Veeresh Sakali	I have mainly worked on the front-end part especially on updating the Profile Page and Sign-up page. Also, helped other teammates working under front-end and hosted weblink.	12.5%
Susmith Meesa	I have worked on class diagram which acts as a blueprint to understand the relation between classes and attributes, I also have assisted in back-end part and fixed the bugs.	12.5%
Harshath Budida	I have worked on Back-end part mainly on GraphQL queries, Mutations and Authentication model and helped in documentation.	12.5%
Harshitha Thokala	I have worked on booking page and Details page for places which comes under front-end part and was a part of presentation making.	12.5%

Gopi Krishna Kummari	I have worked on front-end part Blogs/feedback reactions and Contact form and helped in documentation process.	12.5%
Pooja Sree Poka	I have worked on back-end part mainly on Blogs/Feedback and Contact form and prepared presentation.	12.5%
Sashidhar Chary Viswanathula	I have worked on frontend part which includes login, signup page and also worked on the flow control and communication, prepared the documentations.	12.5%
Balaji Valeti	I have worked on use case diagram which is a successful method for defining and arranging functional needs and documented the test cases.	12.5%