FULLSTACKDEVELOPMENT WITH MERN PROJECT

DOCUMENTATION

**Team Details:**

**Project Title:** [**Flight**](https://github.com/jyothika19/flightfinder-navigating-your-air-travel-options) [**Finder**](https://github.com/jyothika19/flightfinder-navigating-your-air-travel-options)**:** [**Navigating**](https://github.com/jyothika19/flightfinder-navigating-your-air-travel-options) [**your**](https://github.com/jyothika19/flightfinder-navigating-your-air-travel-options) [**air**](https://github.com/jyothika19/flightfinder-navigating-your-air-travel-options) [**travel**](https://github.com/jyothika19/flightfinder-navigating-your-air-travel-options) [**options**](https://github.com/jyothika19/flightfinder-navigating-your-air-travel-options)

**Team Id: LTVIP2025TMID61026**

**Team Members: 4**

1. **Team Leader: SHAIK.SHAN BASHA** – Full Stack Developer &Project Coordinator Responsible for overall planning, coordination, GitHub management, and integration of frontend and backend.
2. **Team Member: BUTI.SANTHOSH**- Frontend Developer Works on the React-based UI, handles component design, page routing, and user interactions.
3. **Team Member: NANAM. ANANTH KUMAR** Backend Developer Builds RESTful APIsusing Node.js and Express.js, manages authentication and server logic.
4. **Team Member**: **ATLURI. SURYA**–Database

Administrator Designs and manages MongoDB schemas, handles CRUD operations and ensures data consistency.

**Flight Booking APP**

# INTRODUCTION

Introducing SB Flights, the ultimate digital platform designed to revolutionize the way you book flight tickets. With SB Flights, your flight travel experience will be elevated to new heights of convenience and efficiency.

Ouruser-friendlywebappempowerstravelersto effortlesslydiscover,explore,andreserveflight tickets based on their unique preferences. Whether you're a frequent commuter or an occasional traveler, finding the perfect flight journey has never been easier.

Imagine accessing comprehensive details about each flight journey at your fingertips. From departureandarrivaltimestoflightclassesandavailableamenities,you'llhavealltheinformation you need to make informed decisions. No more guessing or uncertainty – SB Flightsensures that every aspect of your flight travel is crystal clear.

Thebookingprocessisabreeze.Simplyprovideyourname,age,andpreferredtraveldates,along withthedepartureandarrivalcities,andthenumberofpassengers.Onceyousubmityourbooking request,you'llreceivean instantconfirmationofyourticketreservation.Nomorewaitinginlong queues or dealing with complicated reservation systems – SB Flights makes it quick and hassle- free.

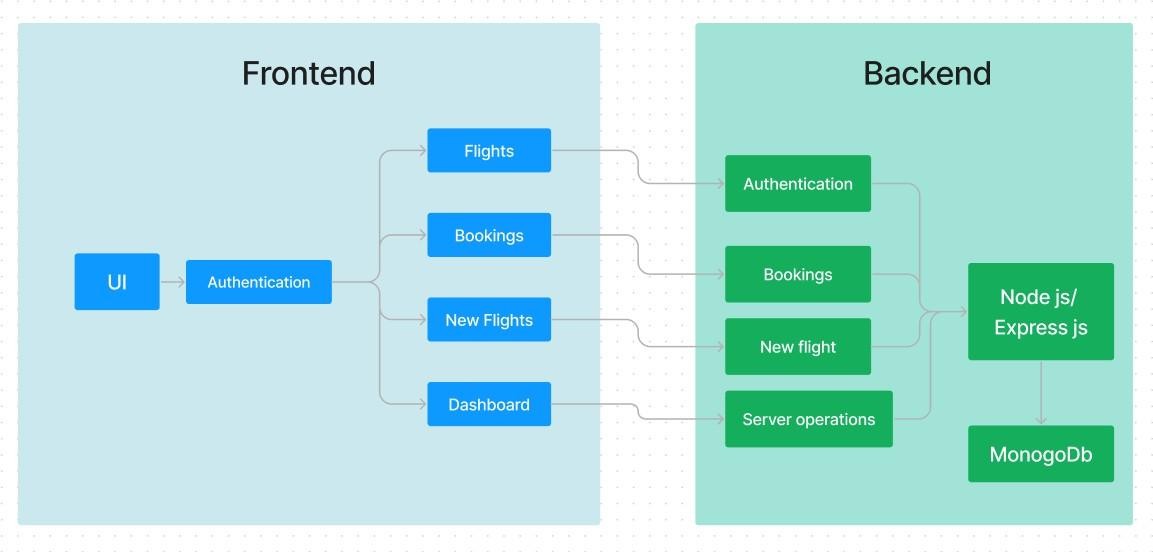
Once your booking is confirmed, our dedicated booking details page becomes your travel companion.Itprovidesacomprehensiveoverviewofyourcurrentandpreviousbookings,allowing you to effortlessly manage your travel plans and stay organized. With SB Flights, you'll have all your essential travel information at your fingertips, ensuring a stress-free journey.

But SB Flights isn't just for travelers. Flight administrators also benefit from our intuitive admin dashboard.Thisspeciallydesigneddashboardempowersadministratorstoefficientlymanageand overseeticketreservationsfortheirflightservice.Theycaneasilyviewthelistofavailableflights for booking and monitor the bookings made by users. With separate login and registration pages for each flight service, privacy and security are always maintained.

SBFlightsisheretoenhanceyourtravelexperiencebyprovidingaseamlessandconvenientway to bookflighttickets.Withouruser-friendlyinterface,efficientbooking management,androbust administrativefeatures,weensureahassle-freeandenjoyableflightticketbookingexperiencefor both users and flight administrators alike.

GetreadytoembarkonaneweraofflighttravelwithSBFlights–yourtickettoeffortlessbooking and unforgettable journeys.

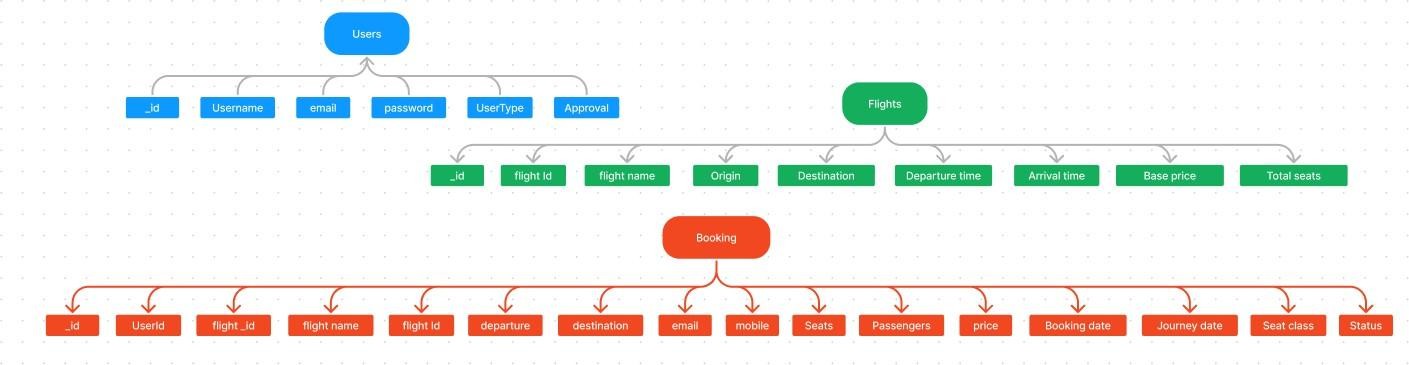
## TECHINICALARCHITECTURE:



Inthisarchitecturediagram:

* The frontend is represented by the "Frontend" section, including user interface components such as User Authentication, Flight Search, and Booking.
* The backend is represented by the "Backend" section, consisting ofAPI endpoints for Users, Flights, Admin and Bookings. It also includes Admin Authentication and an Admin Dashboard.
* The Database section represents the database that stores collections for Users, Flights, and Flight Bookings.

# ERDIAGRAM:

****

TheflightbookingER-diagramrepresentstheentitiesandrelationshipsinvolvedinaflightbooking system. It illustrates how users, bookings, flights, passengers, and payments are interconnected. Here is a breakdown of the entities and their relationships:

**USER:**Representstheindividualsorentitieswhobookflights.Acustomercanplacemultiplebookingsand make multiple payments.

**BOOKING:**Representsaspecificflightbookingmadebyacustomer.Abookingincludesaparticularflight details and passenger information.A customer can have multiple bookings.

**FLIGHT:**Representsaflightthatisavailableforbooking.Here,thedetailsofflightwillbeprovidedandthe users can book them as much as the available seats.

**ADMIN:**Adminisresponsibleforallthebackendactivities.Adminmanagesallthebookings,addsnew flights, etc.,

## Features:

1. **ExtensiveFlightListing:**SBFlightsoffersanextensivelistofflightservices,providingawide range of routes and options for travelers. You can easily browse through the list and explore different flight journeys, including departure and arrival times, flight classes, and available amenities, to find the perfect travel option for your journey.
2. **Book Now Button:** Each flight listing includes a convenient "Book Now" button. When you findaflightjourneythatsuitsyourpreferences,simplyclickonthebuttontoproceedwiththe reservation process.
3. **Booking Details**: Upon clicking the "Book Now" button, you will be directed to a booking details page. Here, you can provide relevant information such as your preferred travel dates, departureandarrivalstations,thenumberofpassengers,andanyspecialrequirementsyoumay have.
4. **Secure and Efficient Booking Process:** SB Flights ensures a secure and efficient booking process.Yourpersonalinformationwillbehandledwiththeutmostcare,andwestrivetomake the reservation process as quick and hassle-free as possible.
5. **Confirmation and Booking Details Page:** Once you have successfully made a reservation, youwillreceiveaconfirmationmessage.Youwillthenberedirectedtoabookingdetailspage, where you can review all the relevant information about your booking, including your travel dates, departure and arrival stations, the number of passengers, and any special requirements you specified.

In addition to these user-facing features, SB Flights provides a powerful admin dashboard, offering administrators a range of functionalities to efficiently manage the system.With the admin dashboard, admins can add and manage multiple flight services, view the list of available flights, monitor user activity, and access booking details for all flight journeys.

SB Flights is designed to enhance your flight travel experience by providing a seamless and user- friendly way to book flight tickets. With our efficient booking process, extensive flight listings, and robust admin dashboard, we ensure a convenient and hassle-free flight ticket booking experience for both users and flight administrators alike.

## PREREQUISITES:

Todevelopafull-stackflightbookingappusingReactJS,Node.js,andMongoDB,there are severalprerequisites you should consider. Hereare the key prerequisites for developing such an application:

**Node.jsandnpm:**InstallNode.js,whichincludesnpm(NodePackageManager),onyour developmentmachine. Node.js is required to run JavaScript on the server side.

* + Download:<https://nodejs.org/en/download/>
  + Installationinstructions:<https://nodejs.org/en/download/package-manager/>

**MongoDB:** Set up a MongoDB database to store hotel and booking information. Install MongoDB locally oruse a cloud-based MongoDB service.

* + Download:<https://www.mongodb.com/try/download/community>
  + Installationinstructions:<https://docs.mongodb.com/manual/installation/>

**Express.js:**Express.jsisawebapplicationframeworkforNode.js.InstallExpress.jstohandleserver- side routing,middleware, and API development.

* + Installation:Openyourcommandpromptorterminalandrunthefollowing command: **npm installexpress**

**React.js**: React.js is a popular JavaScript library for building user interfaces. It enables developers to createinteractive and reusableUI components, making it easierto build dynamicand responsiveweb applications.ToinstallReact.js,aJavaScriptlibraryforbuildinguserinterfaces,followtheinstallation guide: <https://reactjs.org/docs/create-a-new-react-app.html>

**HTML, CSS, and JavaScript:** Basic knowledge of HTML for creating the structure of your app,CSS for styling,and JavaScript for client-side interactivity is essential.

**Database Connectivity:** Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose toconnect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations.

**Front-end Framework:** Utilize Angular to build the user-facing part of the application, includingproductslistings, booking forms, and user interfaces for the admin dashboard.

**Version Control**: Use Git for version control, enabling collaboration and tracking changesthroughoutthedevelopmentprocess.PlatformslikeGitHuborBitbucketcanhost your repository.

* Git:Downloadandinstallationinstructionscanbe foundat:<https://git-scm.com/downloads>

**Development Environment:** Choose a code editor or Integrated Development Environment (IDE) that suits yourpreferences, such as Visual Studio Code, Sublime Text, or WebStorm.

* VisualStudioCode:Downloadfrom<https://code.visualstudio.com/download>
* SublimeText:Downloadfrom<https://www.sublimetext.com/download>
* WebStorm:Downloadfrom<https://www.jetbrains.com/webstorm/download>

### ToConnecttheDatabasewithNodeJSgothroughthebelowprovidedlink:

* Link:<https://www.section.io/engineering-education/nodejs-mongoosejs-mongodb/>

### ToruntheexistingFlightBookingAppprojectdownloadedfromgithub:

Followbelowsteps:

### Clonetherepository:

* Openyourterminalor commandprompt.
* Navigateto thedirectorywhereyouwant to storethee-commerceapp.
* Executethefollowingcommand toclonethe repository:

**Gitclone**:https://github.com/RAFIROCK/flightfinder-navigating-your-air-travel-options

### InstallDependencies:

* Navigateintothecloned repositorydirectory:

### cdFlight-Booking-App-MERN

* Installtherequireddependenciesbyrunningthefollowingcommand:

### npminstall

**StarttheDevelopmentServer:**

* Tostartthedevelopmentserver,executethefollowingcommand:

### npmrundevornpmrun start

* Thee-commerceappwillbeaccessibleat[http://localhost:3000](http://localhost:3000/)bydefault.Youcanchangethe portconfiguration in the .env file if needed.

### AccesstheApp:

* Openyourweb browser and navigateto[http://localhost:3000.](http://localhost:3000/)
* You should see the flight booking app's homepage, indicating that the installation and setup weresuccessful.

Youhavesuccessfullyinstalledandsetuptheflightbookingapponyourlocalmachine.Youcan now proceedwith further customization, development, and testing as needed.

# USER&ADMINFLOW:

### UserFlow:

* + Usersstartbyregistering foran account.
  + Afterregistration,they canlogin withtheir credentials.
  + Once logged in, they can check for the availability of flights in their desired route and dates.
  + Userscanselectaspecificflightfromthe list.
  + Theycanthenproceed byenteringpassengerdetailsandotherrequired data.
  + Afterbooking,theycan viewthedetailsoftheirbooking.

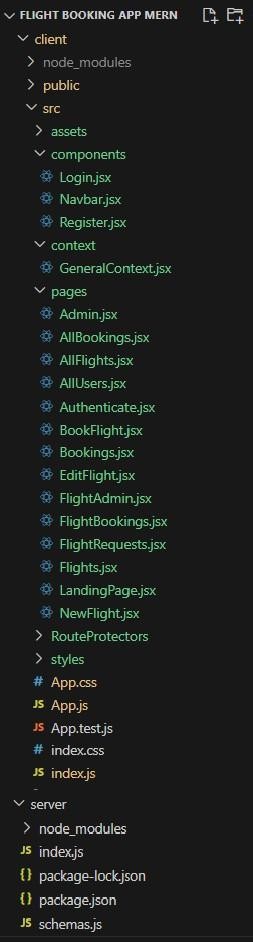
### FlightOperatorFlow:

* + Flightoperatorstartby logginginwiththeir credentials.
  + Onceloggedin,they aredirectedtotheFlightoperator Dashboard.
  + FlightOperatorcanaccesstheDashboard,wheretheycanviewbookings,addnew flight routes, etc.,

### AdminFlow:

* + Adminsstartbylogginginwiththeircredentials.
  + Onceloggedin,theyaredirectedto theAdmin Dashboard.
  + Admins can access theFlight Booking Admin Dashboard,where theycan view bookings, approve new flight operators, etc.,

# PROJECTSTRUCTURE:



This structure assumes a React app and follows a modular approach. Here's a brief explanation of the main directories and files:

* + src/components: Contains components related to the application such as, register, login, home, bookings, etc..
  + src/pageshasthefilesforallthe pagesintheapplication.

# ProjectFlow:

### Milestone1:ProjectSetupand Configuration:

1. **Installrequiredtoolsandsoftware:**
   * Node.js.
   * MongoDB.
   * ReactJs.
   * Git.

### Createprojectfoldersandfiles:

* + Clientfolders.
  + Serverfolders

### Milestone2:BackendDevelopment:

1. **Setupexpressserver:**
   * Installexpress.
   * Createindex.js file.
   * DefineAPI’s

### ConfigureMongoDB:

* + InstallMongoose.
  + Createdatabaseconnection.

### ImplementAPIendpoints:

* + ImplementCRUD operations.
  + TestAPIendpoints.

### Milestone3:WebDevelopment:

1. **SetupReactApplication:**
   * CreateReactappinclient folder.
   * Installrequired libraries
   * Createrequiredpagesandcomponentsandadd routes.

### DesignUIcomponents:

* + CreateComponents.
  + Implementlayoutand styling.
  + Add navigation.

### Implementfrontendlogic:

* + IntegrationwithAPI endpoints.
  + Implementdatabinding.

## Backend:

### SetUpProjectStructure:

* + Createanewdirectoryforyourprojectandsetupapackage.jsonfileusingnpminit command.
  + InstallnecessarydependenciessuchasExpress.js,Mongoose,andotherrequiredpackages.

### DatabaseConfiguration:

* + Set up a MongoDB database either locally or using a cloud-based MongoDB service like MongoDB Atlas or use locally with MongoDB compass.
  + Createadatabaseanddefinethenecessarycollectionsforflights,users,bookings,and other relevant data.

### CreateExpress.jsServer:

* + SetupanExpress.jsservertohandleHTTPrequestsandserveAPIendpoints.
  + Configuremiddlewaresuchasbody-parserforparsingrequestbodiesandcorsfor handling cross-origin requests.

### DefineAPIRoutes:

* + CreateseparateroutefilesfordifferentAPIfunctionalitiessuchasflights,users,bookings, and authentication.
  + Definethenecessaryroutesforlistingflights,handlinguserregistrationandlogin, managing bookings, etc.
  + ImplementroutehandlersusingExpress.jstohandlerequestsandinteractwiththe database.

### ImplementDataModels:

* + DefineMongooseschemasforthedifferentdataentitieslikeflights,users,andbookings.
  + CreatecorrespondingMongoosemodelstointeractwiththeMongoDBdatabase.
  + ImplementCRUDoperations(Create,Read,Update,Delete)foreachmodeltoperform database operations.
  + ​

### UserAuthentication:

* + Createroutes andmiddlewareforuserregistration,login,andlogout.
  + Setupauthenticationmiddlewaretoprotectroutesthatrequireuserauthentication.

### HandlenewFlightsandBookings:

* + Create routes and controllers to handle new flight listings, including fetching flightdata from the database and sending it as a response.
  + Implementbookingfunctionalitybycreatingroutesandcontrollerstohandlebooking requests, including validation and database updates.

### AdminFunctionality:

* + Implementroutesandcontrollersspecifictoadminfunctionalitiessuchasaddingflights, managing user bookings, etc.
  + Addnecessaryauthenticationandauthorizationcheckstoensureonlyauthorizedadmins can access these routes.

### ErrorHandling:

* + Implementerrorhandlingmiddlewaretocatchandhandle anyerrorsthat occurduring the API requests.
  + ReturnappropriateerrorresponseswithrelevanterrormessagesandHTTPstatuscodes.

## Schemausecase:

### UserSchema:

* + Schema: userSchema
  + Model:‘User’
  + TheUserschemarepresentstheuserdataandincludesfieldssuchasusername,email,and password.
  + Itisusedtostoreuserinformationforregistrationandauthenticationpurposes.
  + Theemailfieldismarkedasuniquetoensurethateachuserhasauniqueemail address**.**

### FlightSchema:

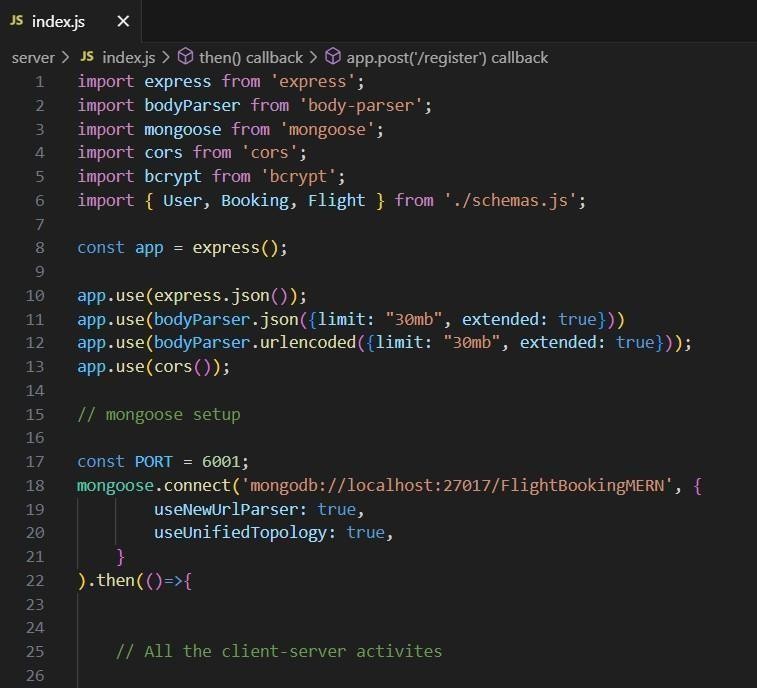
* + Schema: flightSchema
  + Model:‘Flight’
  + TheFlightschemarepresentsthehoteldataandincludesfieldssuchasFlightName, Flight Id, Origin, Destination, Price, seats, etc.,
  + Itisusedtostoreinformationaboutflightsavailableforbookings.

### BookingSchema:

* + Schema: BookingsSchema
  + Model:‘Booking’
  + The Booking schema represents the booking data and includes fields such as userId, flight Name,flightId,Passengers,CoachClass,JourneyDate,etc.,•Itisusedtostoreinformation about the flight bookings made by users.
  + TheuserIdfield isareferencetotheuser whomadethe booking.

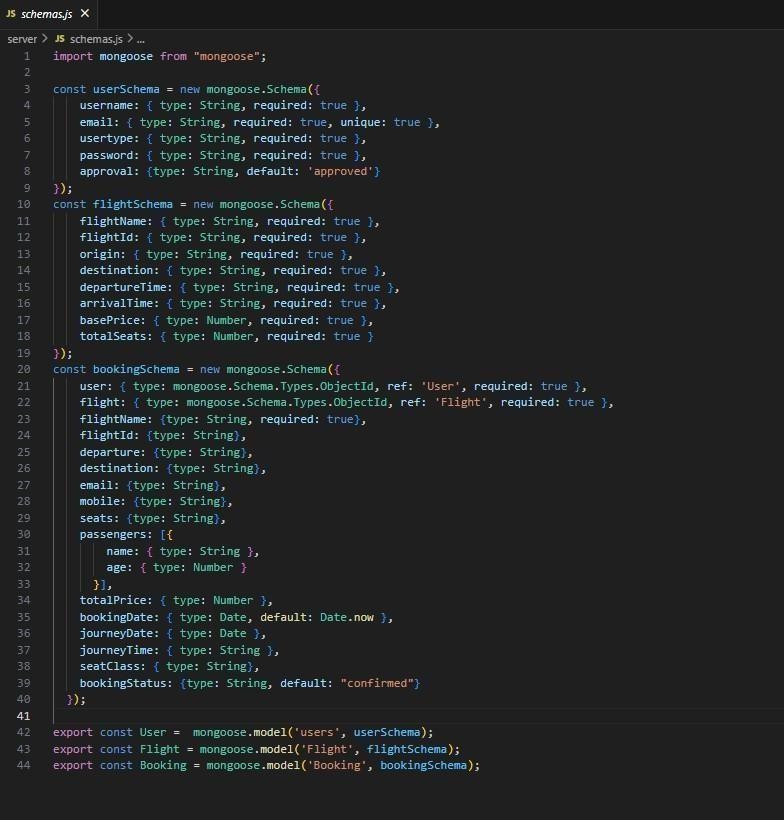
## Code Explanation:

### Serversetup:

Letusimportalltherequiredtools/librariesandconnectthe database.

### Schemas:

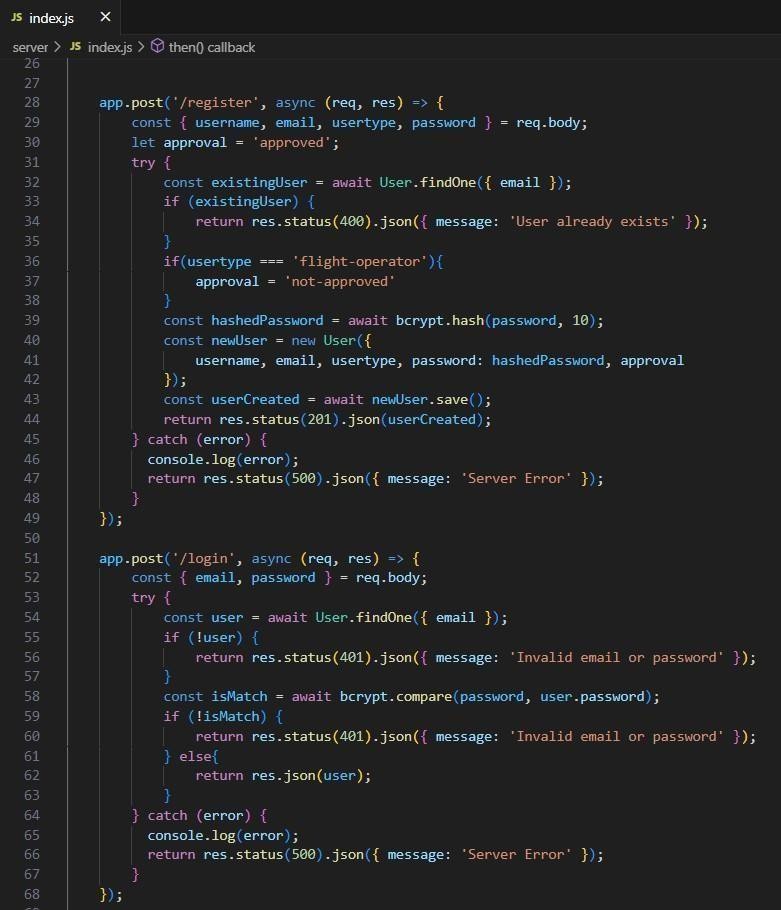
Nowletusdefinetherequiredschemas



### UserAuthentication:

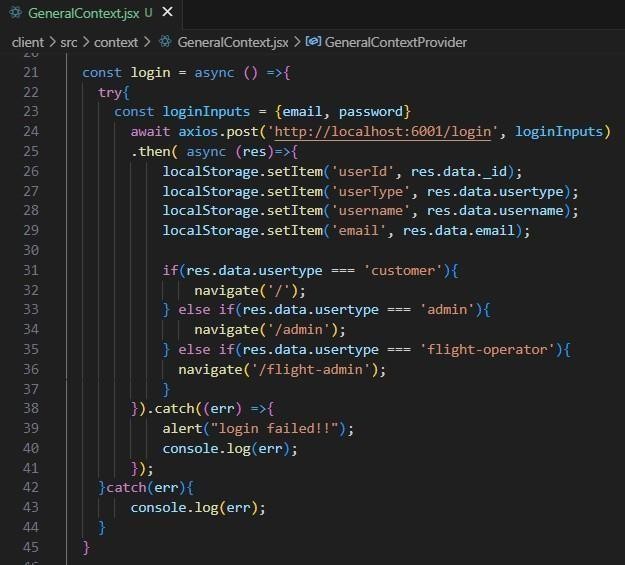
* **Backend**

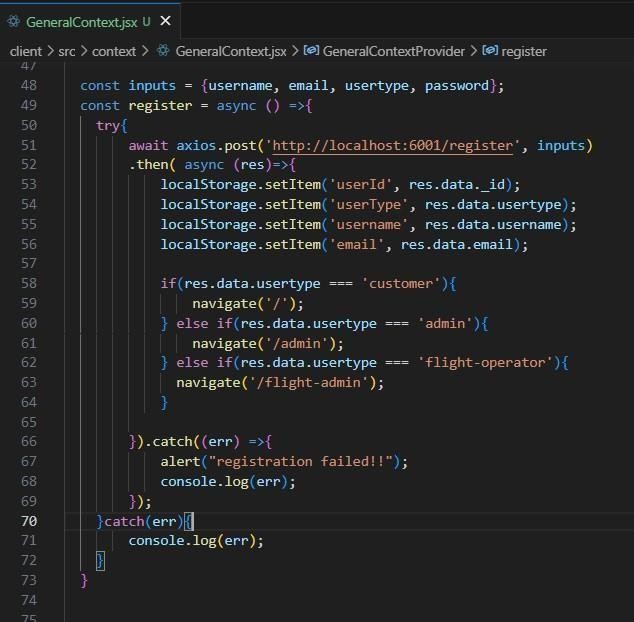
Now,herewedefinethefunctionstohandlehttp requestsfromtheclientfor authentication.

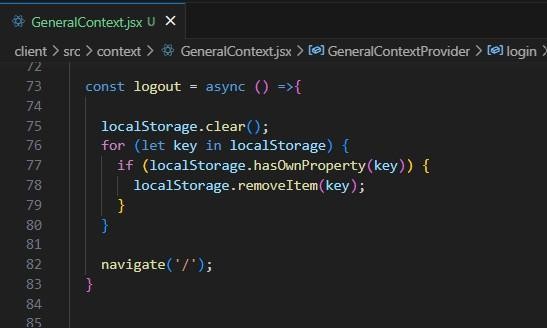


### Frontend

Login:



Register:

Logout:

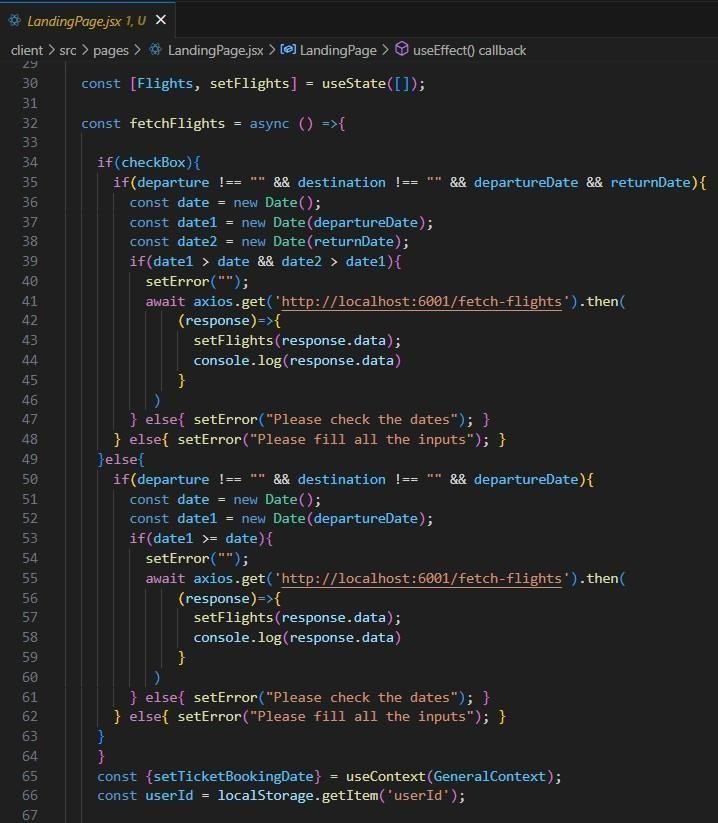
### FlightBooking(User):

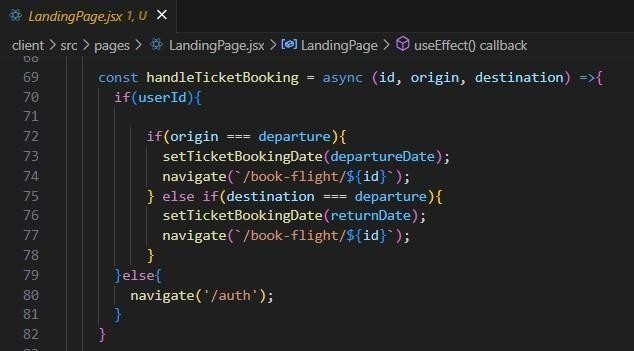
* **Frontend**

Inthefrontend,weimplementedallthebookingcodeinamodal.Initially,weneedto implement flight searching feature with inputs of Departure city, Destination, etc.,

FlightSearchingcode:

Withthegiven inputs,weneedtofetchtheavailableflights.Witheach flight,we adda button to book the flight, which re-directs to the flight-Booking page.



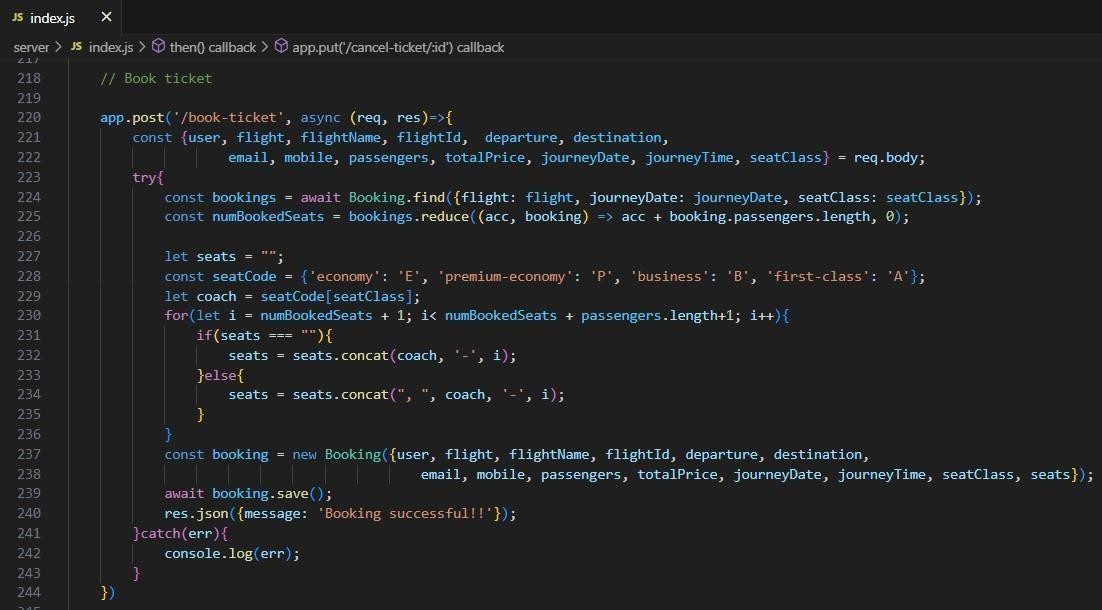
Onselectingthesuitableflight,wethenre-direct totheflight-booking page.

### Backend

Inthebackend,wefetch alltheflightsandthenfilterthemintheclientside.



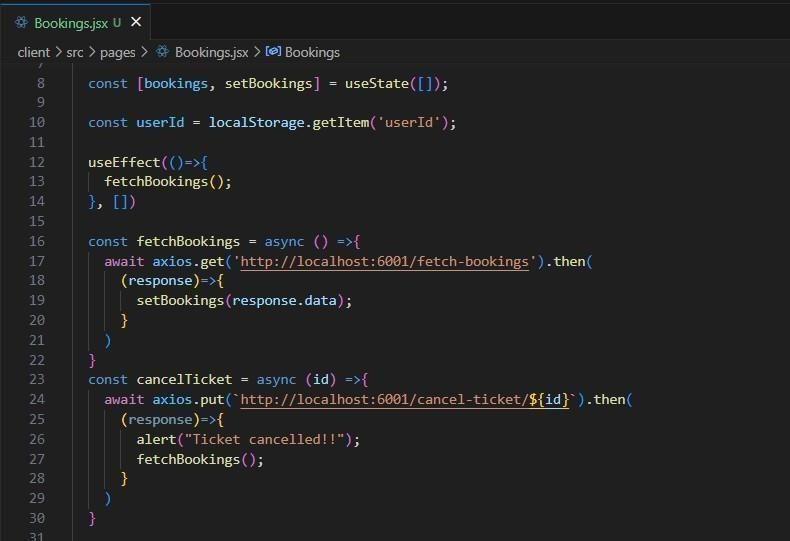
Then,onconfirmation, webookthe flightticketwith theentered details.



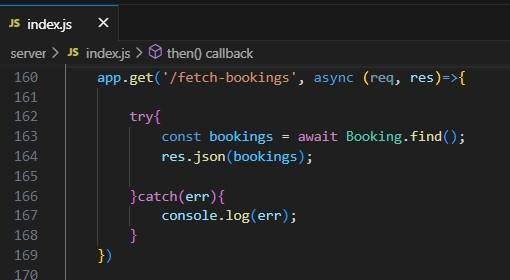
### Fetchinguserbookings:

* **Frontend**

Inthebookingspage,alongwithdisplayingthepastbookings,wewillalsoprovidean option to cancel that booking.



### Backend

Inthebackend,wefetchallthebookingsandthenfilterfortheuser.Otherwise,wecanfetch bookings only for the user.

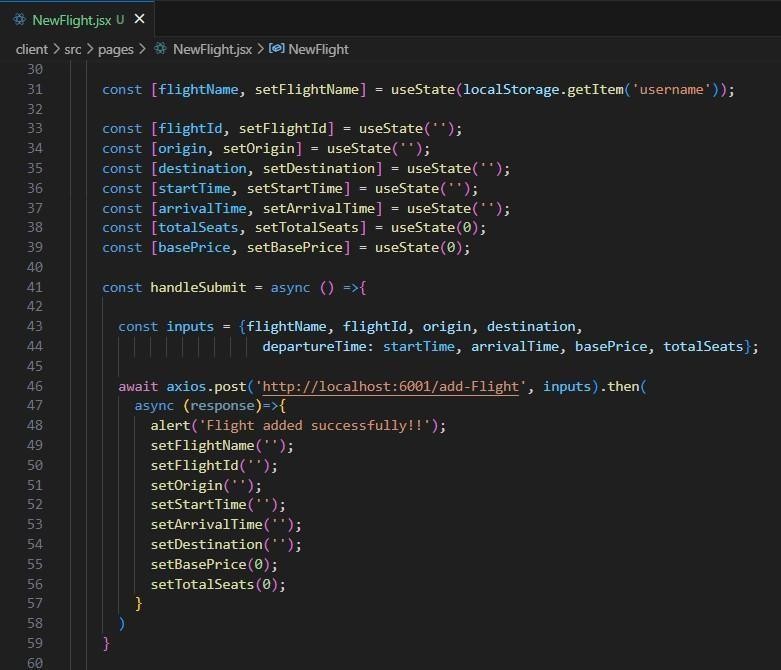
Thenwedefineafunction to delete thebooking on cancelling it on client side.



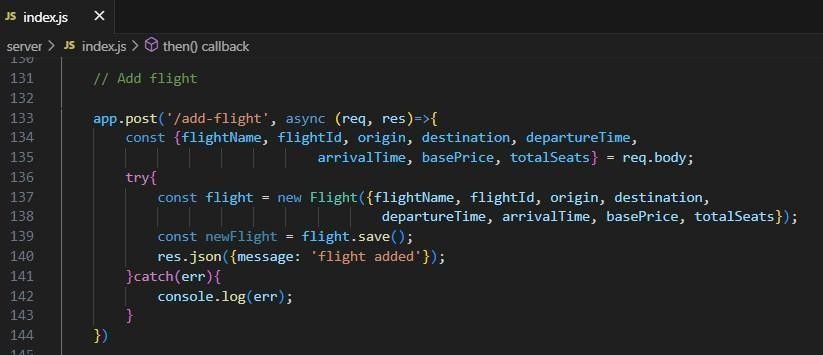
### Addnew flight:

Now,intheadmindashboard,weprovidea functionalitytoaddnewflight.

### Frontend

Wecreateahtmlformwithrequiredinputsforthenewflightandthensendanhttprequest to the server to add it to database.

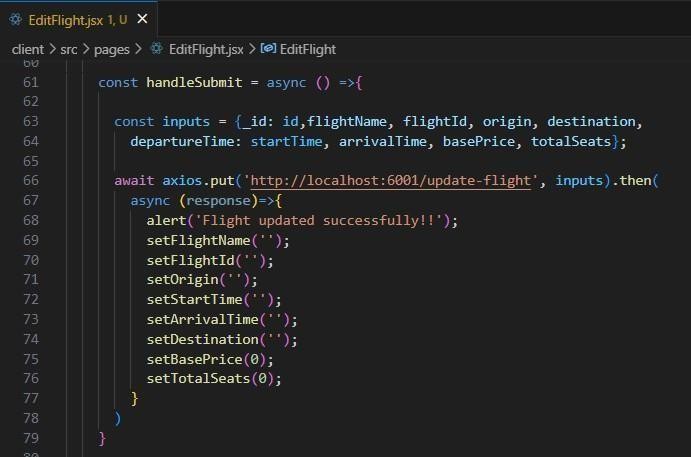
### Backend

In the backend, on receiving the request from the client, we then add the request body to the flight schema.

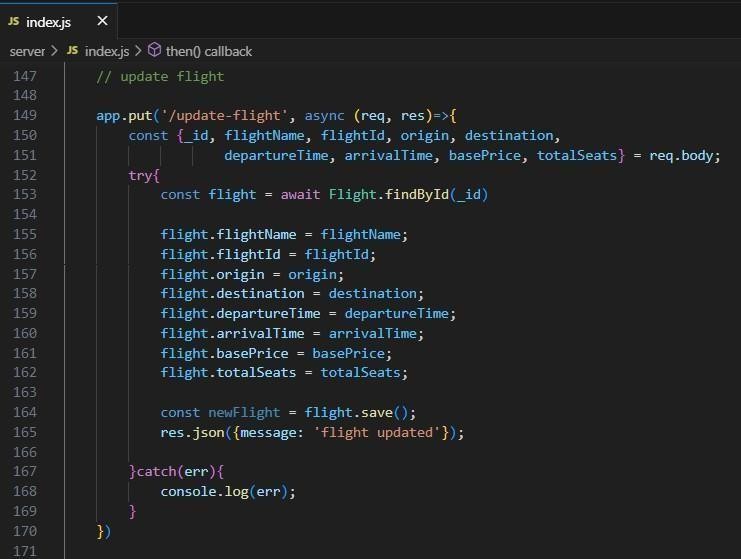
### UpdateFlight:

Here,intheadmindashboard,wewillupdatetheflightdetailsincaseifwe want to make any edits to it

### Frontend:

****

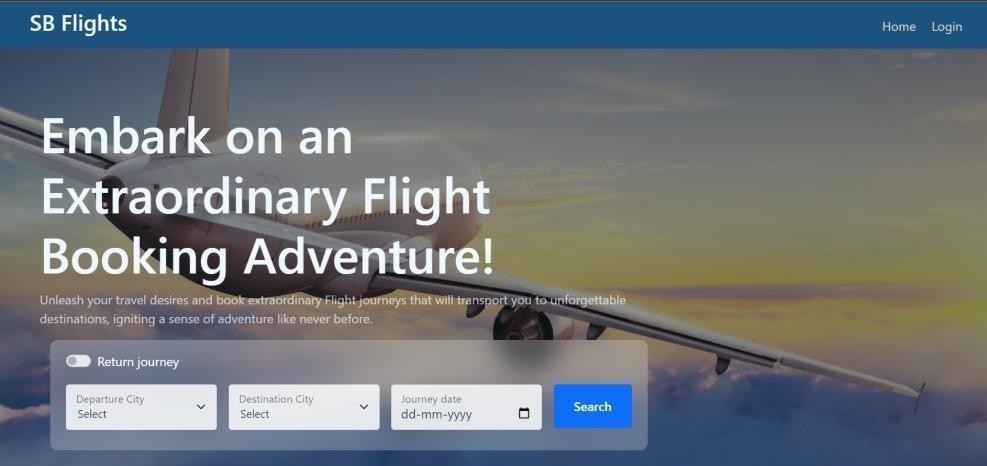
* + **Backend:**



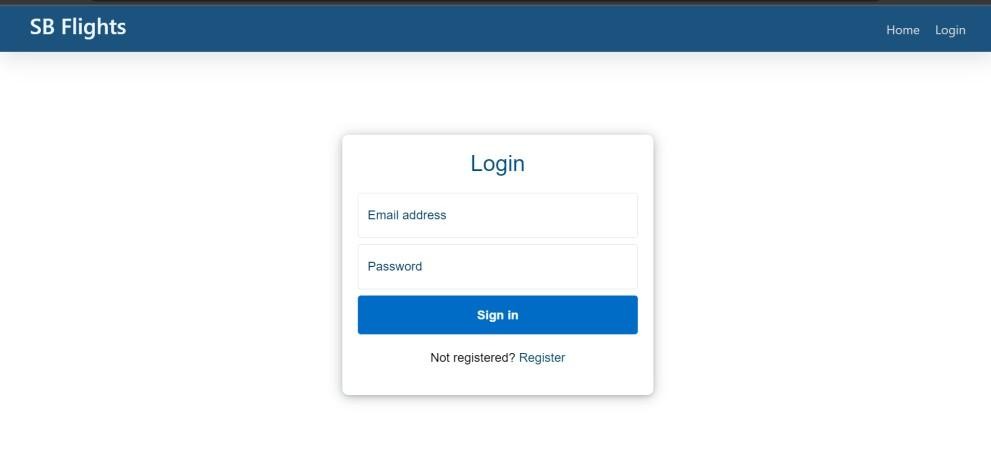
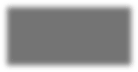
Alongwiththis,implementadditionalfeaturestoviewallflights,bookings,andusersinadmin dashboard.

## DemoUIimages:

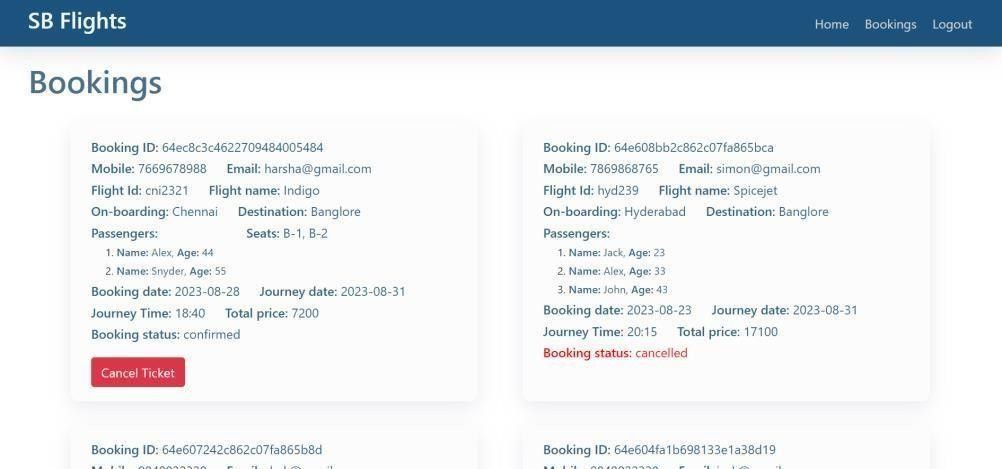
* **Landingpage**



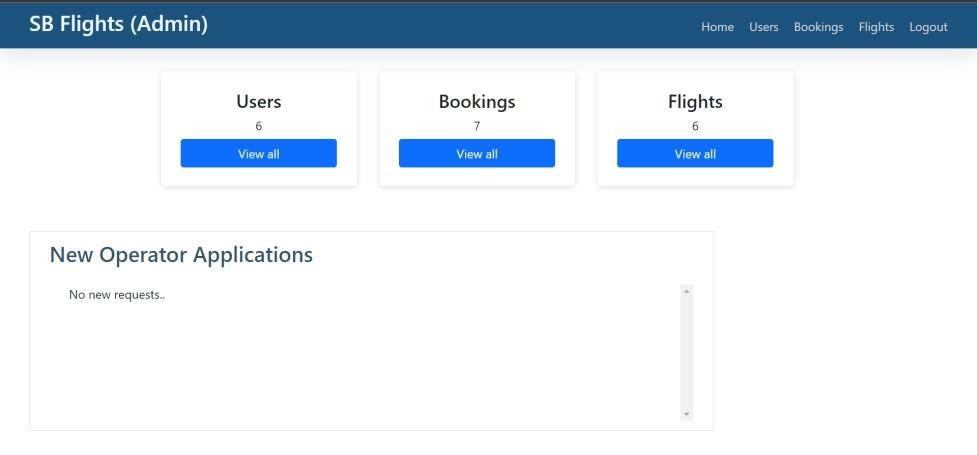
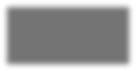
* **Authentication**



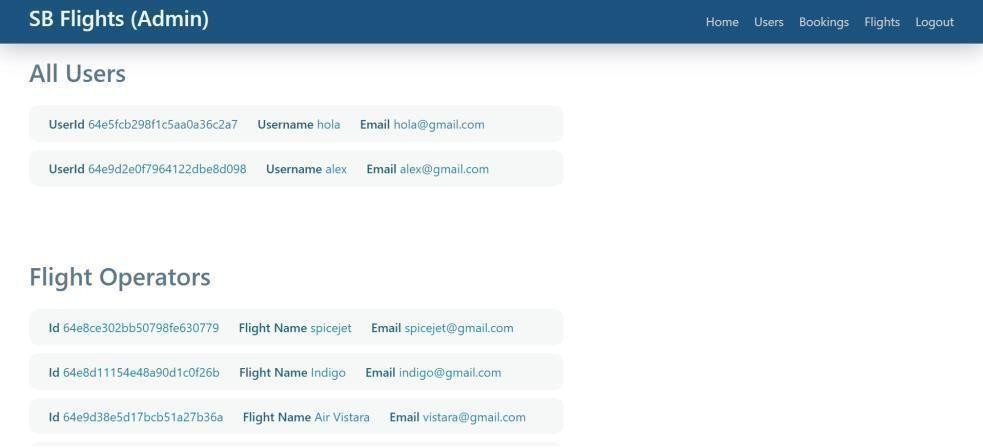
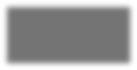
* **Userbookings**



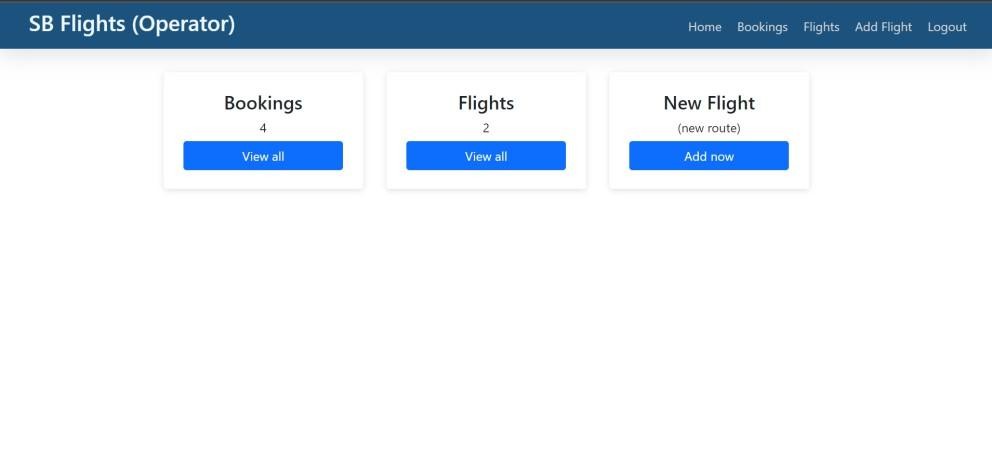
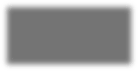
* **AdminDashboard**



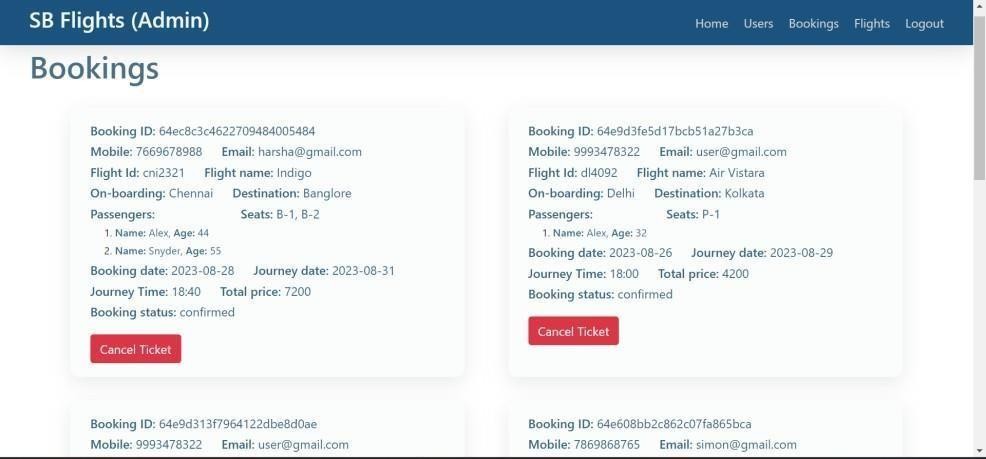
* **All users**



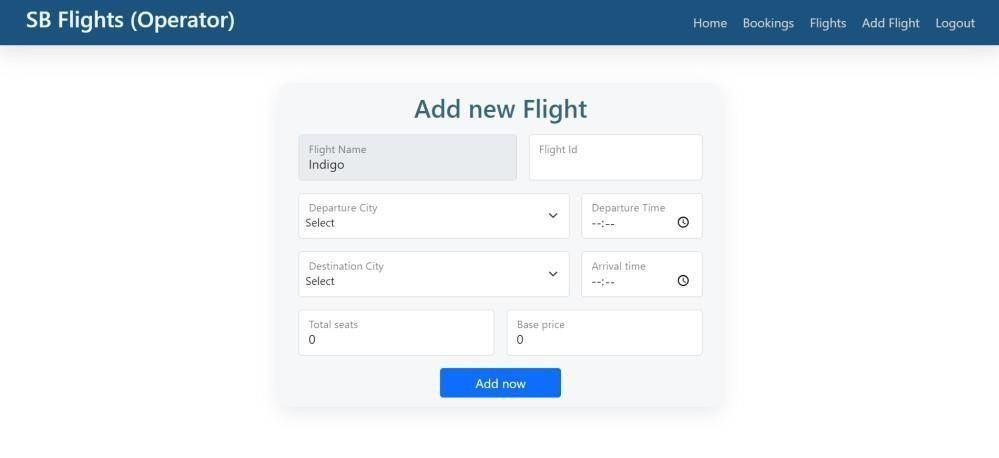
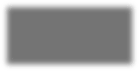
* **FlightOperator**



* **All Bookings**



* **NewFlight**



**-----THANKYOU-----**