

Department of Computer Science

Final Term Project Proposal, summer 20-21

Course Title	ADVANCED PROGRAMMING IN WEB TECHNOLOGY	Section	B
--------------	--	---------	---

Group Members:

Student ID	Name	Signature
18-38712-3	HASAN, MD. NABOBI (Admin)	
18-38784-3	JUBAER, MD. ABU (Employee)	
18-38776-3	MUSTAVI, MUFRAD (Service Provider)	
18-38787-3	SHAHRIAR, MD. TANVIR (User)	

Project Title:

TRAVEL AND TOURISM

Short Description:

Role Name	Description
Admin	Admin can login/logout to the system. If there is no account, can be signed up using google account. Approve hotel accounts, transport accounts. Decline hotel accounts, transport accounts. Remove hotel accounts, transport accounts. View all users/ individual user/ all employees/ individual employee. View active users/employees. Approve new package, tourist places. Decline new package, tourist places. Customize policies. Show income statement. Approve/ Decline salary and festival bonus. Check all boking history. Check all transaction history. Add travel guidelines/employee/ admin. Edit travel guidelines. Delete guidelines/users / employee. Check user review/ support. Can manipulate their profile information (edit-delete-view etc.)
Employee	Employee can login/logout to the system. Apply for new package, tourist places. Edit package, tourist place. Delete package, tourist place. Add discounts. Generate salary. Check salary list, salary status. Calculate festival bonus. Publish income statement. Manage advertisements. Send promotional offers. Check customer account. Manage travel guidelines. Manage FAQ. Check all transaction history. Check review. Manage support. Can manipulate their profile information (edit-delete-view etc.).

Service Provider	Service provider can login/logout to the system. Add hotel room, hotel facilities, car, and flight. Edit hotel room, hotel facilities, car, and flight. Delete hotel room, hotel facilities, car, and flight. Update availability of individual car, flight and room. Approve Booking of hotel, car and flight. Decline Booking of hotel, car and flight. Show all booking details of hotel, car and flight. Show booked users details. Manage car and flight type. Manage pricing for different type of car and flight. Check number of rooms, flights and cars. Check all cars, flights and hotel room. Support. Check reviews. Check transaction report. Can manipulate their profile information (edit-delete-view etc.).
User	User can login/logout to the system. Explore tourist places. Search travel packages. Explore Hotels. Choose Hotel rooms. Book bus, flight. Book hotel. Check Booking list. Hotel cancelation request. Bus and flight ticket cancelation request. Check travel guidelines. Show Travel History. Notifications. Support. About us. FAQ. Give Review. Can manipulate their profile information (edit-delete-view etc.).

Modules:

Login Module (M1)		
Function	Description	Database
M1F1	There will be a login page with 2 input fields one will take the username and another will take input of the user's password. There will also be a login button which will be used to submit form data for successful login to the system using a post request. There is another hyperlink "Not yet Registered" which will redirect to registration page using a get request and there is another button for signup with google account, which will allow to sign up as an Admin.	N/A
M1F2	<p>A backend form validation will occur which will validate the user's credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none">• Null username and password field validation.• Username length < 20 characters• Password length min. 4 characters and max. 20 characters. <p>If any invalid input is given then error message should be shown to the view page accordingly.</p>	N/A
M1F3	Upon passing the validation steps the user must be searched by the username in the database and get the users username and password. Then the password must be checked with the user submitted password if it matches with the database password for validation. If it matches then necessary credentials such as user type, username, id, image should be saved in session storage and redirected to the home page of the user according to the user type using get request. Upon not finding the user from database or password mismatch an error message must be shown to the view.	Search from database table "Users", "Admins", "Employees", "Transports", "Hotels" and column "username" and return "password", "username", "id", "type" fields.

Logout Module (M2)		
Function	Description	Database
M2F1	There has logout link in system header after clicking logout button the authentication session will remove and user redirect to M1F1 .	N/A

Registration Module (M3)		
Function	Description	Database
M3F1	In the view of Module M1F1 a “Not yet Registered” hyperlink will be redirect to the registration type selection page which will contain three hyperlink naming “User Registration”, “Hotel Service” and “Transport Service”. This hyperlink will redirect serially to user registration page, hotel registration page and transport registration page using get request.	N/A
M3F2	After clicking “User Registration” from M3F1 , in the view there’ll be a html form having fields First Name, Last Name, Choose Gender, Email, Phone, Username, Password, Re-type Password and a submit button to fire form submission to the same route with post request.	N/A
M3F3	<p>A backend form validation will occur which will validate the user’s credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Null First Name, Last Name, Gender, Email, Phone, Username, Password and Re-Type password field validation. • First Name length min. 2 characters and max. 10 characters. • Last Name length min. 2 characters and max. 10 characters. • Email format checking, length max. 30 characters, Uniqueness checking. • Phone length min. 11 characters and max. 15 characters. • Password length min. 4 characters and max. 20 characters. • Re-Type Password length min. 4 characters and max. 20 characters. <p>For any validation error an error message will be shown. For successful validation the user will be recorded to the database with a registration completion message.</p>	<p>Match data from database table “Users” and column “email”, “username”.</p> <p>Store firstname = firstname, lastname = lastname, gender=gender, email=email, username= username, password=password, phone=phone, type= “User”, status= “Active” to respective columns of table “Users”.</p>
M3F4	After clicking “Hotel Service” from M3F1 , in the view there’ll be a html form having fields Name, Location, Address, Phone Number, Email, Password, Re-Type Password, Hotel Image and a submit button to fire form submission to the same route with post request.	N/A

<p>M3F5</p>	<p>A backend form validation will occur which will validate the user's credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Name, Location, Address, Phone Number, Email, Password, Re-Type Password, Hotel Image field validation. • Name length min. 2 characters and max. 20 characters. • Location length min. 2 characters and max. 20 characters. • Email format checking, length max. 30 characters, Uniqueness checking. • Phone length min. 11 characters and max. 15 characters. • Password length min. 4 characters and max. 20 characters. • Re-Type Password length min. 4 characters and max. 20 characters, must be same as Password. <p>For any validation error an error message will be shown. For successful validation the hotel account will be recorded to the database with a registration completion message.</p>	<p>Match data from database table "Hotels" and column "email", "name".</p> <p>Store name = name, location= location, address= address, email=email, phone=phone, password=password, image=image, type= "Hotel", request= "Pending" to respective columns of table "Hotels".</p>
<p>M3F6</p>	<p>After clicking "Transport Service" from M3F1, in the view there'll be a html form having fields Service Provider, Name, Phone Number, Email, Password, Re-Type Password and a submit button to fire form submission to the same route with post request.</p>	<p>N/A</p>
<p>M3F7</p>	<p>A backend form validation will occur which will validate the user's credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Service Provider, Name, Phone Number, Email, Password, Re-Type Password field validation. • Name length min. 2 characters and max. 20 characters. • Email format checking, length max. 10 characters, Uniqueness checking. • Phone length min. 11 characters and max. 15 characters. • Password length min. 4 characters and max. 20 characters. • Re-Type Password length min. 4 characters and max. 20 characters, must be same as Password. <p>For any validation error an error message will be shown. For successful validation the hotel account will be recorded to the database with a registration completion message.</p>	<p>Match data from database table "Transports" and column "email", "name".</p> <p>Store name = name, email=email, phone=phone, password=password, type= Service Provider, request= "Pending" to respective columns of table "Transports".</p>

Navigation bar and header Module-Admin (M4)		
Function	Description	Database
M4F1	<p>Admin's header will show current date and contain 2 hyperlinks:</p> <ul style="list-style-type: none"> Profile: "/admin/profile" (GET Request) Logout: "/logout" (GET Request) 	N/A
M4F2	<p>Admin's navigation bar will contain a picture section for showing admin's image some hyperlinks to navigate to various channels such as:</p> <ul style="list-style-type: none"> Dashboard: "/admin" Admin <ul style="list-style-type: none"> Add Admin: "/admin/add-admin" (GET Request) Admin List: "/admin/all-admins" (GET Request) User <ul style="list-style-type: none"> User List: "/admin/all-users" (GET Request) Active User List: "/admin/active-users" (GET Request) Employee <ul style="list-style-type: none"> Add Employee: "/admin/add-employee" (GET Request) Employee List: "/admin/all-employees" (GET Request) Active Employee List: "/admin/active-employees" (GET Request) Hotel <ul style="list-style-type: none"> Approve/Decline: "/admin/hotels-pending" (GET Request) All Hotels: "/admin/all-hotels" (GET Request) 	Select image from table "admins" and column "image" for showing the admins image.

	<ul style="list-style-type: none"> ○ Hotel Bookings: “/admin/hotel-bookings” (GET Request) ● Transport <ul style="list-style-type: none"> ○ Approve/Decline: “/admin/transport-pending” (GET Request) ○ All Car Companies: “/admin/all-cars” (GET Request) ○ All Flight Companies: “/admin/all-flight” (GET Request) ○ Car Bookings: “/admin/car-bookings” (GET Request) ○ Flight Bookings: “/admin/flight-bookings” (GET Request) ● Packages <ul style="list-style-type: none"> ○ Approve /Delete: “/admin/package-pending” (GET Request) ○ All Packages: “/admin/all-packages” (GET Request) ○ Package Bookings: “/admin/package-bookings” (GET Request) ○ Package Status: “/admin/package-status” (GET Request) ● Places <ul style="list-style-type: none"> ○ Approve/Decline: “/admin/place-pending” (GET Request) ○ All Places: “/admin/all-places” (GET Request) ● Accounts <ul style="list-style-type: none"> ○ Income Statement: “/admin/income-statement” (GET Request) ○ Transaction History: “/admin/transaction-history” (GET Request) ○ Approve/Decline Salary: 	
--	---	--

	<p>“/admin/pending-salary” (GET Request)</p> <ul style="list-style-type: none"> • Support: “/admin/support” (GET Request) • Policies: “/admin/policy” (GET Request) • Guidelines: “/admin/guidelines” (GET Request) • About Us: “/admin/about” (GET Request) 	
M4F3	<p>Admin’s navigation bar will contain a picture section for showing admin’s image some hyperlinks to navigate to various channels such as:</p> <ul style="list-style-type: none"> • Dashboard: “/admin” • Admin <ul style="list-style-type: none"> ○ Add Admin: “/admin/add-admin” (GET Request) ○ Admin List: “/admin/all-admins” (GET Request) • User <ul style="list-style-type: none"> ○ User List: “/admin/all-users” (GET Request) ○ Active User List: “/admin/active-users” (GET Request) • Employee <ul style="list-style-type: none"> ○ Add Employee: “/admin/add-employee” (GET Request) ○ Employee List: “/admin/all-employees” (GET Request) ○ Active Employee List: “/admin/active-employees” (GET Request) • Hotel <ul style="list-style-type: none"> ○ Approve/Decline: “/admin/hotels-pending” (GET Request) 	<p>Select image from table “admins” and column “image” for showing the admins image.</p>

	<ul style="list-style-type: none"> ○ All Hotels: “/admin/all-hotels” (GET Request) ○ Hotel Bookings: “/admin/hotel-bookings” (GET Request) ● Transport <ul style="list-style-type: none"> ○ Approve/Decline: “/admin/transport-pending” (GET Request) ○ All Car Companies: “/admin/all-cars” (GET Request) ○ All Flight Companies: “/admin/all-flight” (GET Request) ○ Car Bookings: “/admin/car-bookings” (GET Request) ○ Flight Bookings: “/admin/flight-bookings” (GET Request) ● Packages <ul style="list-style-type: none"> ○ Approve /Delete: “/admin/package-pending” (GET Request) ○ All Packages: “/admin/all-packages” (GET Request) ○ Package Bookings: “/admin/package-bookings” (GET Request) ○ Package Status: “/admin/package-status” (GET Request) ● Places <ul style="list-style-type: none"> ○ Approve/Decline: “/admin/place-pending” (GET Request) ○ All Places: “/admin/all-places” (GET Request) ● Accounts <ul style="list-style-type: none"> ○ Income Statement: “/admin/income-statement” (GET Request) ○ Transaction History: “/admin/transaction-history” (GET Request) 	
--	---	--

	<ul style="list-style-type: none"> ○ Approve/Decline Salary: “/admin/pending-salary” (GET Request) • Support: “/admin/support” (GET Request) • Policies: “/admin/policy” (GET Request) • Guidelines: “/admin/guidelines” (GET Request) • About Us: “/admin/about” (GET Request) 	
--	--	--

Navigation bar and Header Module - Employee (M5)		
Function	Description	Database
M5F1	<p>Employee’s header will contain 2 hyperlinks:</p> <ul style="list-style-type: none"> • Profile “/employee/profile” • Logout “/logout” 	N/A
M5F2	<p>Employee’s navigation bar will have several channels:</p> <ul style="list-style-type: none"> • Dashboard “/employee” • Users “/employee/userlist” • Packages <ul style="list-style-type: none"> • Add Package “/employee/packageAdd” • Manage Package “/employee/package” • Places <ul style="list-style-type: none"> • Add Place “/employee/placeAdd” • Manage Place “/employee/place” 	N/A

	<ul style="list-style-type: none"> • Gallery <ul style="list-style-type: none"> • Add Image “/employee/gallery” • Manage Album “/employee/galleryManage • Salary <ul style="list-style-type: none"> • Generate Salary “/employee/salary” • Salary List “/employee/salary/list” • Accounts <ul style="list-style-type: none"> • Add Income Statement “/employee/statementAdd” • Income Statement “/employee/statement” • Add in Transaction History “/employee/transactionAdd” • Transaction History “/employee/transaction” • Advertisement <ul style="list-style-type: none"> • Advertisement “/employee/advertisement” • Send Promo “/employee/promo” • Review “employee/Review” • Support “employee/support” • Guidelines “employee/guideline” • FAQ “employee/faq” 	
--	--	--

Navigation & Header Module(M6) – Service Provider (Hotel)		
Function	Description	Database
M6F1	<p>Hotel header will contain 2 channels:</p> <p>Profile: “/hotelDashboard/profile”</p> <p>Logout: “/logout”</p>	N/A
M6F2	<p>Hotel navigation bar will contain some hyperlinks which will navigates to various channels such as:</p> <ul style="list-style-type: none"> • Home: “/hotelDashboard” • Facilities: <ul style="list-style-type: none"> ➤ New Facilities: “/hotelDashboard/addhotelfacility” ➤ Manage Facilities: “/hotelDashboard/managehotelfacility” • New Room: <ul style="list-style-type: none"> ➤ Add Room: “/hotelDashboard/addhotelroom” ➤ Manage Room: “/hotelDashboard/managehotelroom” • Booking: <ul style="list-style-type: none"> ➤ Booking List: “/hotelDashboard/showhotelallboking” ➤ Manage Booking: “/hotelDashboard/room-book-pending” • Check Review: “/hotelDashboard/checkhotelreview” • Transaction History: “/hotelDashboard/hoteltransactionhistory” • Support: “/hotelDashboard/hotelsupport” 	N/A

Navigation & Header Module(M7) – Service Provider (Flight)		
Function	Description	Database
M7F1	<p>Flight header will contain 2 channels:</p> <p>Profile: “/flightDashboard/profile”</p> <p>Logout: “/logout”</p>	N/A
M7F2	<p>Flight navigation bar will contain some hyperlinks which will navigates to various channels such as:</p> <ul style="list-style-type: none"> • Home: “/flightDashboard” • New Airplane: <ul style="list-style-type: none"> ➤ Add Airplane: “/flightDashboard /addflight” ➤ Manage Airplane: “/flightDashboard /manageflight” • Flight Type: “/flightDashboard/flighttype” • Booking: <ul style="list-style-type: none"> ➤ Booking List: “/flightDashboard/flight-booking-list” ➤ Manage Booking: “/flightDashboard/flight-pending-list” • Availability: “flightDashboard/flightavailability” • Check Review: “/flightDashboard / /checkflightreview” • Transaction History: “/flightDashboard / flighttransactionhistory” • Support: “/flightDashboard / flightsupport” 	N/A

Navigation & Header Module(M8) – Service Provider (Car)		
Function	Description	Database
M8F1	<p>Car header will contain 2 channels:</p> <p>Profile: “/carDashboard/profile”</p> <p>Logout: “/logout”</p>	N/A
M8F2	<p>Car navigation bar will contain some hyperlinks which will navigates to various channels such as:</p> <ul style="list-style-type: none"> • Home: “/carDashboard” • New Car: <ul style="list-style-type: none"> ➤ Add Car: “/carDashboard/addcar” ➤ Manage Car: “/carDashboard/managecar” • Car Type: “/carDashboard/cartype” • Booking: <ul style="list-style-type: none"> ➤ Booking List: “/carDashboard/car-booking-list” ➤ Manage Booking: “/carDashboard/car-pending-list” • Availability: “/carDashboard/caravailability” • Check Review: “/carDashboard/checkcarreview” • Transaction History: “/carDashboard/cartransactionhistory” • Support: “/carDashboard/carsupport” 	N/A

Header Module (M9)		
Function	Description	Database
M9F1	<p>The header will contain some hyperlinks to navigate to various channels and the following hyperlinks:</p> <ul style="list-style-type: none"> • Home: “/user” (Get Request) • Booking: <ul style="list-style-type: none"> ○ Flight: “/user/flight” (Get Request) ○ Hotels: “/user/hotel” (Get Request) ○ Car: “/user/car” (Get Request) • Service: <ul style="list-style-type: none"> ○ Destination: “/user/destination” (Get Request) ○ Packages: “/user/package” (Get Request) ○ Hotels: “/user/show_hotels” (Get Request) ○ Flight: “/user/flight_list” (Get Request) ○ Car: “/user/Car_list” (Get Request) • Travel History: “/user/travel_history” (Get Request) • Travel Guideline: “/user/guideline” (Get Request) • Notification: “/user/notification” (Get Request) • About: <ul style="list-style-type: none"> ○ About Us: “/user/about” (Get Request) ○ Contact Us: “/user/contact” (Get Request) ○ Privacy Policy: “/user/privay_policy” (Get Request) ○ FAQ: “/user/faq” (Get Request) • Review: “/user/give_review” (Get Request) • Profile: “/user/profile” (Get Request) • LogOut: “/logout” (Get Request) 	N/A

Dashboard Module-Admin (M10)		
Function	Description	Database
M10F1	<p>A dashboard channel should be developed in the system in the route “/admin” where multiple card section will be available. It will also contain M4F1 in the top and M4F2 in the left side of the view page.</p> <p>1st card will show the number of active users and a hyperlink named “More Info” which redirects to “/admin/all-users”.</p> <p>2nd card will show the number of active employees and a hyperlink named “More Info” which redirects to “/admin/all-employees”.</p> <p>3rd card will show the number of messages in support and a hyperlink named “More Info” which redirects to “/admin/support”.</p> <p>4th card will show the number of hotel bookings and a hyperlink named “More Info” which redirects to “/admin/hotel-bookings”.</p> <p>5th card will show the number of car bookings and a hyperlink named “More Info” which redirects to “/admin/car-bookings”.</p> <p>6th card will show the number of flight bookings and a hyperlink named “More Info” which redirects to “/admin/flight-bookings”.</p> <p>7th card will show the number of upcoming packages and a hyperlink named “More Info” which redirects to “/admin/all-packages”.</p>	<p>Select count of active users from table “users” and column “status”.</p> <p>Select count of active employees from table “employees” and column “status”.</p> <p>Select count of messages from table “supports” and column “status”.</p> <p>Select count of hotel bookings from table “roombooks” and column “req”.</p> <p>Select count of car bookings from table “carbooks” and column “req”.</p> <p>Select count of flight bookings from table “airbooks” and column “req”.</p> <p>Select count of upcoming packages from table “packages” and column “req”.</p>

Dashboard Module - Employee (M11)		
Function	Description	Database
M11F1	<p>In the view of M11F1 the modules of M5F1 and M5F2 will be added.</p> <p>There will be 6 cards containing different channels.</p> <p>The Users card shows the number of users and the More Info button redirects to “/employee/userList”</p> <p>The Package card shows the number of users and the More Info button redirects to “/employee/package”</p> <p>The Places card shows the number of users and the More Info button redirects to “/employee/place”</p> <p>The Hotels card shows the number of users and the More Info button redirects to “/employee/hotelList”</p> <p>The Transport card shows the number of users and the</p>	<p>In Users the rows of database table “users” will be shown.</p> <p>In Packages the rows of database table “packages” will be shown where req=“Approved”.</p> <p>In Places the rows of database table “places” will be shown where req=“Approved”.</p> <p>In Hotels the rows of database table “hotels” will be shown where req=“</p>

	<p>More Info button redirects to “/employee/transportList”</p> <p>The Supports card shows the number of users and the More Info button redirects to “/employee/support”</p>	<p>Approved”.</p> <p>In Transports the rows of database table “transports” will be shown where req=”Approved”.</p> <p>In Supports the rows of database table “supports” will be shown.</p>
--	--	--

Dashboard Module(M12) – Service Provider (Hotel)		
Function	Description	Database
M12F1	On the home view page on route “/hotelDashboard” there would be a dashboard which shows number total rooms, number of booked rooms and the number of reviews.	<p>Search hotel_id and count number of total rooms from db table “rooms”.</p> <p>Search hotel_id and count number of booked rooms from db table “roombooks”.</p> <p>Search hotel_id and count number of reviews from db table “reviews”.</p>
M12F2	<p>Total rooms, booked rooms and reviews in the dashboard, they have a hyperlink named ‘More Info’ which will redirect them to the particular page which are:</p> <p>Total Rooms: “/hotelDashboard/managehotelroom”</p> <p>Booked Rooms: “/hotelDashboard/showhotelallboking”</p> <p>Review: “/hotelDashboard/checkhotelreview”</p>	N/A

Dashboard Module(M13) – Service Provider (Flight)		
Function	Description	Database
M13F1	On the home view page on route “/flightDashboard” there would be a dashboard which shows number of total airs, number of pending booking and the number of reviews.	<p>Search flight_id and count number of total airs from db table “airs”.</p> <p>Search flight_id and count number of pending booking from db table “airbooks”.</p> <p>Search flight_id and count number of reviews from db table “reviews”.</p>
M13F2	<p>Total airs, number of pending bookings and reviews in the dashboard, they have a hyperlink named ‘More Info’ which will redirect them to the particular page which are:</p> <p>Total Airs: “/flightDashboard/manageflight”</p> <p>Pending Booking: “/flightDashboard/flight-pending-list”</p> <p>Review: “/flightDashboard/checkflightreview”</p>	N/A

Dashboard Module(M14) – Service Provider (Car)		
Function	Description	Database
M14F1	On the home view page on route “/carDashboard” there would be a dashboard which shows number of total cars, number of pending booking and the number of reviews.	<p>Search car_id and count number of total car from db table “cars”.</p> <p>Search car_id and count number of pending booking from db table “carbooks”.</p> <p>Search car_id and count number of reviews from db table “reviews”.</p>

M14F2	<p>Total cars, number of pending bookings and reviews in the dashboard, they have a hyperlink named 'More Info' which will redirect them to the particular page which are:</p> <p>Total Car: "/carDashboard/managecar"</p> <p>Pending Booking: "/carDashboard/car-pending-list"</p> <p>Review: "/carDashboard/checkcarreview"</p>	N/A
--------------	---	-----

Profile Module-Admin (M15)		
Function	Description	Database
M15F1	<p>A profile channel should be developed in the system in the route "/admin/profile" where there'll be a html form having fields First Name, Last Name, Choose Gender, Email, Username, Password, Re-type Password and 2 submit buttons to fire form submission to the same route with post request. One Button for update profile and another for delete profile.</p>	N/A
M15F2	<p>A backend form validation will occur which will validate the admin's credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Null Email, Username field validation. • Re-Type password requires with Password • Email format checking, length max. 30 characters, Uniqueness checking. • Username length min. 3 characters. <p>For any validation error an error message will be shown. For successful validation the admin account will be updated (If update button clicked) to the database with an update completion message or account will be deleted (If delete button clicked) from database and redirect to M1F1.</p>	<p>Match data from database table "admins" and column "email", "username"</p> <p>Find admin from database table "admins" by "id"</p> <p>Updates firstname=firstname, lastname=lastname, gender=gender, email=email, Username=username, password=password</p>

Profile Module-Employee (M16)		
Function	Description	Database
M16F1	A profile channel should be developed in the system in the route “/employee/profile” where there’ll be a html form having fields First Name, Last Name, Choose Gender, Email, Username, Password, Re-type Password and 2 submit buttons to fire form submission to the same route with post request. One Button for update profile and another for deactivate profile.	N/A
M16F2	<p>A backend form validation will occur which will validate the employee’s credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Null Email, Username field validation. • Re-Type password requires with Password • Email format checking, length max. 30 characters, Uniqueness checking. • Username length min. 3 characters. <p>For any validation error an error message will be shown. For successful validation the employee account will be updated (If update button clicked) to the database with an update completion message</p>	<p>Match data from database table “employees” and column “email”, “username”</p> <p>Find admin from database table “admins” by “id”</p> <p>Updates firstname=firstname, lastname=lastname, gender=gender, email=email, Username=username, password=password</p>
M16F3	If deactivate button clicked account will be deactivated from database and redirect to M1F1 with a delete completion message.	In the database table “ employees ” “ status ” = Deactivated will be updated.

Profile Module (M17)- Service Provider (Hotel)		
Function	Description	Database
M17F1	A profile channel should be developed in the system in the route “/hotelDashboard/profile” where there’ll be a html form having fields Hotel Name, Hotel Location, Hotel Address, Phone, Email, Type, Password, Re-type Password and 2 submit buttons to fire form submission to the same route with post request. One Button for Update profile and another for Delete profile.	N/A
M17F2	<p>A backend form validation will occur which will validate the user’s credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Null Email, name field validation. • Re-Type password requires with Password • Email format checking, length max. 30 characters, Uniqueness checking. • name length min. 3 characters. <p>For any validation error an error message will be shown. For successful validation the flight account will be</p>	<p>Match data from database table “hotels” and column “email”, “name”</p> <p>Find hotel from database table “hotels” by “id”</p> <p>Update’s name=name, location=location, address=address, phone =phone, email=e mail, type=type,</p>

	updated (If update button clicked) to the database with an update completion message or account will be deleted (If delete button clicked) from database and redirect to M17F1 .	password=password
--	---	--------------------------

Profile Module (M18)- Service Provider (Flight)		
Function	Description	Database
M18F1	A profile channel should be developed in the system in the route “/flightDashboard/profile” where there’ll be a html form having fields Company Name, Company Phone, Company Email, Company Type, Password, Re-type Password and 2 submit buttons to fire form submission to the same route with post request. One Button for Update profile and another for Delete profile.	N/A
M18F2	<p>A backend form validation will occur which will validate the user’s credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Null Email, Username field validation. • Re-Type password requires with Password • Email format checking, length max. 30 characters, Uniqueness checking. • name length min. 3 characters. <p>For any validation error an error message will be shown. For successful validation the flight account will be updated (If update button clicked) to the database with an update completion message or account will be deleted (If delete button clicked) from database and redirect to M18F1.</p>	<p>Match data from database table “transports” and column “email”, “name”</p> <p>Find flight from database table “transports” by “id”</p> <p>Update’s name=name, phone =phone, email=e mail, type=type, password=password</p>

Profile Module (M19)- Service Provider (Car)		
Function	Description	Database
M19F1	A profile channel should be developed in the system in the route “/carDashboard/profile” where there’ll be a html form having fields Company Name, Company Phone, Company Email, Company Type, Password, Re-type Password and 2 submit buttons to fire form submission to the same route with post request. One Button for Update profile and another for Delete profile.	N/A

M19F2	<p>A backend form validation will occur which will validate the user's credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Null Email, name field validation. • Re-Type password requires with Password • Email format checking, length max. 30 characters, Uniqueness checking. • name length min. 3 characters. <p>For any validation error an error message will be shown. For successful validation the flight account will be updated (If update button clicked) to the database with an update completion message or account will be deleted (If delete button clicked) from database and redirect to M19F1.</p>	<p>Match data from database table "transports" and column "email", "name"</p> <p>Find flight from database table "transports" by "id"</p> <p>Update's name=name, phone =phone, email=e mail, type=type, password=password</p>
--------------	---	---

Profile Module-User (M20)		
Function	Description	Database
M20F1	<p>A profile channel should be developed in the system in the route "/user/profile" where there'll be a html form having fields First Name, Last Name, Gender, Email, Phone, Username, Password, Confirm Password and 2 submit buttons to fire form submission to the same route with post request. One Button for update profile and another for delete profile.</p>	N/A
M20F2	<p>A backend form validation will occur which will validate the user's credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Null Email, Username field validation. • Re-Type password requires with Password • Email format checking, length max. 30 characters, Uniqueness checking. • Username length min. 3 characters. <p>For any validation error an error message will be shown. For successful validation the admin account will be updated (If update button clicked) to the database with an update completion message or account will be Deactivated (If delete button clicked) from database and redirect to M1F1.</p>	<p>Match data from database table "users" and column "email", "username"</p> <p>Updates firstname=firstname, lastname=lastname, gender=gender, email=email, phone=phone, username=username, password=password.</p>

Admin Management Module (M21)		
Function	Description	Database
M21F1	An admin add channel should be developed in the system in the route “/admin/add-admin” where there’ll be a html form having fields First Name, Last Name, Choose Gender, Email, Add Image Username, Password, Re-type Password and a submit buttons to fire form submission to the same route with post request.	N/A
M21F2	<p>A backend form validation will occur which will validate the admin’s credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <p>Null First Name, Last Name, Choose Gender, Email, Add Image Username, Password, Re-type Password field validation.</p> <p>FirstName length min. 2 characters and max. 20 characters.</p> <p>LastName length min. 2 characters and max. 20 characters.</p> <p>Email format checking, length max. 30 characters, Uniqueness checking.</p> <p>Username length min. 3 characters. Uniqueness checking.</p> <p>Password length min. 4 characters and max. 20 characters.</p> <p>Re-Type password requires with Password and same as Password</p> <p>For any validation error an error message will be shown.</p> <p>For successful validation the admin account will be created to the database with an adding completion message.</p>	<p>Match database from table “admins” and column “email”, “username”.</p> <p>Store firstname = firstname, lastname = lastname, gender= gender, email=email, image=image, username=username, password=password, type= “Admin” to respective columns of table “admins”.</p>
M21F3	<p>An admin list channel should be developed in the system in the route “/admin/all-admins” where there’ll be a html table having columns ID, Username , First Name, Last Name, Gender, Image, Email and Action.</p> <p>Contents of action column in each row will be referring the route “/admin/admin/delete/{id}”.</p>	<p>Select all data from table “admins”</p> <p>Of database’</p> <p>Uses the value of “id” column of the database table “admins”</p>
M21F4	<p>On clicking delete hyperlink beside each admin on M21F3 it would create a “GET” request to “/admin/admin/delete/{id}” with the id of that particular admin.</p> <p>An admin delete channel should be developed in the system in the route “/admin/admin/delete/{id}” where there’ll be a html table having columns ID, Username, First Name, Last Name, Gender, Email and a confirm button which will create a post request to “/admin/admin/delete/{id}”.</p> <p>It would delete the row of the admin of the table “admins” And redirect to M21F3 with a get request.</p>	<p>Delete a row from table “admins” by using id = “id”</p>

User Management Module-Admin (M22)		
Function	Description	Database
M22F1	<p>A user list channel should be developed in the system in the route “/admin/all-user” where there’ll be a html table having columns ID, Username, First Name, Last Name, Gender, Email, Status and Action. Contents of action column in each row will have two hyperlinks:</p> <ul style="list-style-type: none"> • User Details Referring the route “/admin/user/details/{id}”. • User Delete Referring the route “/admin/user/delete/{id}”. 	<p>Select all data from table “users” Of database’</p> <p>Uses the value of “id” column of the database table “users”</p>
M22F2	<p>On clicking details hyperlink beside each admin on M22F1 it would create a “GET” request to “/admin/user/details/{id}” with the id of that particular user.</p> <p>A user details channel should be developed in the system in the route “/admin/user/details/{id}” where there’ll be a html table having columns ID, Username, First Name, Last Name, Gender and Email</p>	<p>Select data from table “users” of database, by using id = “id”</p>
M22F3	<p>On clicking delete hyperlink beside each user on M22F1 it would create a “GET” request to “/admin/user/delete/{id}” with the id of that particular user.</p> <p>A user delete channel should be developed in the system in the route “/admin/user/delete/{id}” where there’ll be a html table having columns ID, Username, First Name, Last Name, Gender, Email and a confirm button which will create a post request to “/admin/user/delete/{id}”. It would delete the row of the user of the table “users” And redirect to M22F1 with a get request.</p>	<p>Delete a row from table “users” by using id = “id”</p>
M22F4	<p>An active user list channel should be developed in the system in the route “/admin/active-user” where there will be a html table having columns ID, Username, First Name, Last Name, Gender, Email and Status. Only active users will be shown here in this table.</p>	<p>Select all data from table “users” of database where status = ‘Active’</p>

Show User Module - Employee (M23)		
Function	Description	Database
M23F1	<p>In the view of M23F1 the modules of M5F1 and M5F2 will be added. Also there will be a table containing Id, Firstname, Lastname, Gender, Details column which will show the userlist.</p> <p>Details column will have details button which will hold the id of that row's user.</p>	All the rows from database table " users " will be shown with " id ", " firstname ", " lastname ", " gender " columns.
M23F2	<p>The details option holding id parameter will open a new page routing "/employee/userlist{id}". In that page, the "id", "firstname", "lastname", "gender". "email", "phone" will be shown for that user.</p> <p>There will be a back button which will redirect to the previous page.</p>	Find users from database " users " with " id " and show the " firstname ", " lastname ", " gender ", " email ", " phone " columns for that user.

Home Page Module (M24)		
Function	Description	Database
M24F1	<p>After clicking "LOGIN" button from M1F1 it will redirect to the user's home page route "/user".</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. The view page contains a search button which makes post request to "/user" and it is used for searching hotels. Beside the search button, there are two hyperlinks for finding flights and cars. The "Find Flight" hyperlink redirects to "/user/flight_list" and "Find Car" hyperlink redirects to "/user/Car_list". The view page also contains "View All Places" hyperlink for choosing destinations and the hyperlink redirect to "/user/destination". The page also contains hyperlinks for "Packages" which redirects to "/user/package". And for "Hotels" the hyperlinks redirect to "/user/show_hotels".</p>	N/A
M24F2	<p>After clicking "Find Flight" hyperlink from M24F1 it will redirect to "/user/flight_list".</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be M38F1 which will contain the necessary contents.</p>	Select all data from table " airs " where availability="Available" of database.
M24F3	<p>After clicking "Find Car" hyperlink from M24F1 it will redirect to "/user/Car_list".</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be M43F1 which will contain the necessary contents.</p>	Select all data from table " cars " where availability="Available" of database.

M24F4	<p>Clicking the hyperlink named “Search Hotel” would create a “GET” request to “/user/show_hotels” with the hotel of that particular district.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html view page matching the column “location” from M32F1 having columns Hotel ID, Hotel Name, Hotel Location, Hotel Address, Phone Number and Email.</p>	<p>Find hotel from database table “hotels” by “location”.</p> <p>Select all data from table “hotels” where req= “Approved” of database.</p>
--------------	---	---

Destination Module (M25)		
Function	Description	Database
M25F1	<p>A destination channel should be developed in the system in the route “/user/destination”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html view page having Destination ID, Place, District Image. With the district name there will be a hyperlink which will redirect to “/user/destination_details{id}”.</p>	<p>Select all data from table “places” where req= “Approved” of database.</p>
M25F2	<p>Clicking the hyperlink from M25F1 would create a “GET” request to “/user/destination_details{id}” with the id of that particular place.</p> <p>A destination details channel should be developed in the system in the route “/user/destination_details{id}”, where there’ll be an html table having column Id, Place, District, Image and Hotel. Under the “Hotel” column there will be a hyperlink which will redirect to “/user/search_hotels{district}”.</p>	<p>Find place from database table “places” by “id”.</p> <p>Select all data from table “places” where req= “Approved” of database.</p>
M25F3	<p>Clicking the hyperlink from M25F2 would create a “GET” request to “/user/search_hotels{district}” with the district of that particular place.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html view page matching the column “district” from M25F1 with the column “location” from M32F1 having Hotel ID, Hotel Name, Hotel Location, Hotel Address, Phone Number and Email.</p>	<p>Find hotel from database table “hotels” by “location”.</p> <p>Select all data from table “hotels” where req= “Approved” of database.</p>

Employee Management Module-Admin (M26)		
Function	Description	Database
M26F1	An employee add channel should be developed in the system in the route “/admin/add-employee” where there’ll be a html form having fields First Name, Last Name, Choose Gender, Email, Username, Password, Re-type Password and a submit buttons to fire form submission to the same route with post request.	N/A
M26F2	<p>A backend form validation will occur which will validate the employee’s credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <p>Null First Name, Last Name, Choose Gender, Email, Username, Password, Re-type Password field validation. FirstName length min. 2 characters and max. 20 characters. LastName length min. 2 characters and max. 20 characters. Email format checking, length max. 30 characters, Uniqueness checking. Username length min. 3 characters. Uniqueness checking. Password length min. 4 characters and max. 20 characters. Re-Type password requires with Password and same as Password</p> <p>For any validation error an error message will be shown. For successful validation the employee account will be created to the database with an adding completion message.</p>	<p>Match database from table “employees” and column “email”, “username”.</p> <p>Store firstname = firstname, lastname = lastname, gender=gender, email=email, username= username, password=password, type = “Employee”, status = “Active” to respective columns of table “employees”.</p>
M26F3	<p>An employee list channel should be developed in the system in the route “/admin/all-employees” where there’ll be a html table having columns ID, Username, First Name, Last Name, Gender, Image, Email, Status and Action.</p> <p>Contents of action column in each row will have two hyperlinks:</p> <ul style="list-style-type: none"> Employee Details Referring the route “/admin/employee/details/{id}”. Employee Delete Referring the route “/admin/employee/delete/{id}”. 	<p>Select all data from table “employees” of database’</p> <p>Uses the value of “id” column of the database table “employees”</p>
M26F4	<p>On clicking details hyperlink beside each employee on M26F1 it would create a “GET” request to “/admin/employee/details/{id}” with the id of that particular employee.</p> <p>An employee details channel should be developed in the system in the route “/admin/user/details/{id}” where</p>	<p>Select data from table “employees” of database, by using id = “id”</p>

	there'll be a html table having columns ID, Username, First Name, Last Name, Gender, Email and Status	
M26F5	<p>On clicking delete hyperlink beside each employee on M26F3 it would create a "GET" request to "/admin/employee/delete/{id}" with the id of that particular admin.</p> <p>An employee delete channel should be developed in the system in the route "/admin/employee/delete/{id}" where there'll be a html table having columns ID, Username, First Name, Last Name, Gender, Email, Status and a confirm button which will create a post request to "/admin/employee/delete/{id}". It would delete the row of the employee of the table "employees" And redirect to M26F3 with a get request.</p>	Delete a row from table "employees" by using id = "id"
M26F6	<p>An active employee list channel should be developed in the system in the route "/admin/active-employee" where there will be a html table having columns ID, Username, First Name, Last Name, Gender, Email and Status. Only active employees will be shown here in this table.</p>	Select all data from table "employees" of database, where status = 'Active'

Gallery Module - Employee (M27)		
Function	Description	Database
M27F1	<p>In the view the modules M5F1 and M5F2 will be added and there will be a form with a "Upload" button which will refer to the route "/employee/gallery"</p> <p>In the view a html form will contain "Place", "image" options.</p>	N/A
M27F2	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> Place length must be less than 50 characters. Image will be file type but stores as string in database. 	Store "place" = "place", "image" = "image" in the galleries table of database.
M27F3	<p>In the view of this modules of M5F1 and M5F2 will be added.</p> <p>A Manage Place channel should be developed in the system in the route "/employee/galleryManage" where there'll be a html table having columns Place, Image and Action. Contents of action column in each row will be parameterized with an id , will delete the row.</p>	Select id="id" from database table galleries

M27F4	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> • Place length must be less than 50 characters. • Image will be file type but stores as string in database. 	Delete entire row from database table galleries where id="id"
--------------	---	---

Hotel Management Module-Admin (M28)		
Function	Description	Database
M28F1	<p>A pending hotel list channel should be developed in the system in the route “/admin/hotels-pending” where there’ll be an html table having columns ID, Name, Address, Image, Contact No., Email and Action. Contents of action column in each row will have two hyperlinks:</p> <ul style="list-style-type: none"> • Approve Referring the route “/admin/hotel/approve{id}”. • Decline Referring the route “/admin/hotel/decline{id}”. 	<p>Select all data from table “employees” of database’ Where req = “Pending”</p> <p>Uses the value of “id” column of the database table “hotels”</p>
M28F2	<p>On clicking approve hyperlink beside each hotel on M28F1 it would create a “GET” request to “/admin/hotel/approve{id}” with the id of that particular hotel.</p> <p>An approve hotel channel should be developed in the system in the route “/admin/hotel/approve{id}” where there’ll be a html table having columns Id, Name, Location, Address, Image, Contact No., Email and a confirm button which will create a post request to “/admin/ hotel/approve/{id}”.</p> <p>It would update the row of the hotel of the table “hotels” Column req = “Approved” And redirect to M28F1 with a get request.</p>	<p>Find hotel from database table “hotels” by “id”</p> <p>Updates req= “Approved”</p>
M28F3	<p>On clicking decline hyperlink beside each hotel on M28F1 it would create a “GET” request to “/admin/hotel/decline{id}” with the id of that particular hotel.</p> <p>A decline hotel channel should be developed in the system in the route “/admin/hotel/decline{id}” where there’ll be a html table having columns Id, Name, Location, Address, Image, Contact No., Email and a confirm button which will create a post request to “/admin/ hotel/decline/{id}”.</p> <p>It would delete the row of the hotel of the table “hotels” And redirect to M28F1 with a get request.</p>	Delete a row from table “hotels” by using id = “id”

M28F4	<p>A hotel list channel should be developed in the system in the route “/admin/all-hotels” where there’ll be a html table having columns ID, Name, Location, Address, Image, Contact No., Email and Action. Contents of action column in each row will be referring the route “/admin/hotel/delete/{id}”.</p>	<p>Select data from table “hotels” of database, where req = “Approved”</p> <p>Uses the value of “id” column of the database table “hotels”</p>
M28F5	<p>On clicking delete hyperlink beside each hotel on M28F4 it would create a “GET” request to “/admin/hotel/delete/{id}” with the id of that particular hotel.</p> <p>A hotel account delete channel should be developed in the system in the route “/admin/hotel/delete/{id}” where there’ll be a html table having columns ID, Name, Location, Address, Image, Contact No., Email and a confirm button which will create a post request to “/admin/hotel/delete/{id}”.</p> <p>It would delete the row of the hotel of the table “hotels” And redirect to M28F4 with a get request.</p>	<p>Delete a row from table “hotels” by using id = “id”</p>
M28F6	<p>A hotel booking list channel should be developed in the system in the route “/admin/hotels-bookings” where there’ll be a html table having columns Id, User Id, Room Id, Arrival Date, Departure Date, Status and Action. Contents of action column in each row will be referring the route “/admin/hotel/ hotels-bookings/{id}”.</p>	<p>Select data from table “roombooks” of database</p>

Facilities Module(M29) – Service Provider (Hotel)		
Function	Description	Database
M29F1	<p>Facilities module has two hyperlinks which will navigates to channels such as:</p> <ul style="list-style-type: none"> New Facilities: “/hotelDashboard/addhotelfacility” Manage Facilities: “/hotelDashboard/managehotelfacility” 	N/A
M29F2	<p>In the M29F1, New Facilities view page there’ll be a html form having fields add image, facility title, description and a add button to fire form submission to the same route with post request.</p>	N/A

M29F3	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • image: Required • title: Required, length min. 2 • description: Required, length min. 2 <p>If everything is fine image, title, description will be stored in the database. After then user will redirect to M29F2.</p>	Store's image = image, title=title, description = description to the db table " facilities "
M29F4	<p>In the M29F1, Manage Facilities view page there will be a table having column S.no, Facility Title, Description, Image, Creation date. All the respective data will be shown from database.</p> <p>Action column will be use to show action buttons such as Edit and Delete.</p>	List will use db table " facilities " by hotel_id and columns id, title, description, image, created .
M29F5	In the List of Module M29F4 Edit Button will be added to every row referring the route " /hotelDashboard/facilityedit{id} ". Here id parameter holds the hotel id of a row.	Uses the value of " id " column of the db table " facilities "
M29F6	In the view there'll be a html form having fields facility title, description and a update button to fire form submission to the same route with post request.	N/A
M29F7	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • title: Required, length min. 2 • description: Required, length min. 2 <p>If everything is fine title, description will be stored in the database. After then user will redirect to M29F4.</p>	<p>Find title and description from db table "facilities" by "id"</p> <p>Store's title=title, description= description</p>
M29F8	In the List of Module M29F4 a Delete Button will be added to every row referring the route " /hotelDashboard/facilitydelete{id} ". Here id parameter holds the id of a row.	Uses the value of " id " column of the db table " facilities "
M29F9	In the view there'll be a confirmation. After confirm a virtual form will be submitted to the route of M29F8 on Delete Request.	N/A
M29F10	After receiving delete request, a particular facility will be retrieved from the database and will be removed. After then user will redirect to M29F4 .	<p>Find id, facility title, description, creation date from db table "facilities" by "id"</p> <p>Delete from db table "facilities" by "id"</p>

Room Module(M30) – Service Provider (Hotel)		
Function	Description	Database
M30F1	<p>Room module has two hyperlinks which will navigates to channels such as:</p> <ul style="list-style-type: none"> Add Room: “/hotelDashboard/addhotelroom” Manage Room: “/hotelDashboard/managehotelroom” 	N/A
M30F2	In the M30F1 , Add Room view page there'll be a html form having fields room name, room price, room description, room image and a add button to fire form submission to the same route with post request.	N/A
M30F3	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> name: Required, length min. 2 price: Required, length min. 2 and max. 15 description: Required, length min. 2 image: Required <p>If everything is fine name, price, description, image will be stored in the database. After then user will redirect to M30F2.</p>	Store's name = name, image = image, price = price, description = description to the db table “rooms”
M30F4	<p>In the M30F1, Manage Rooms view page there will be a table having column S.no, Room Name, Description, Image, Room Price, Room Availability, Creation date. All the respective data will be shown from database.</p> <p>Action column will be use to show action buttons such as Edit and Delete.</p>	List will use db table “rooms” by hotel_id and columns id, name, description, image, price, created, availability .
M30F5	In the List of Module M30F4 an Edit Button will be added to every row referring the route “/hotelDashboard/roomedit{id}”. Here id parameter holds the hotel id of a row.	Uses the value of “ id ” column of the db table “rooms”
M30F6	In the view there'll be a html form having fields room name, room price, room description, room availability and a update button to fire form submission to the same route with post request.	N/A

M30F7	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • name: Required, length min. 2 • price: Required, length min. 2 and max. 9 • description: Required, length min. 2 • availability: Required <p>If everything is fine name, price, description, availability will be stored in the database. After then user will redirect to M30F4.</p>	<p>Find name, price, availability and description from db table “rooms” by “id”</p> <p>Store’s name=name, description= description, price=price, availability=availability</p>
M30F8	<p>In the List of Module M30F4 a Delete Button will be added to every row referring the route “/hotelDashboard/roomdelete{id}”. Here id parameter holds the id of a row.</p>	<p>Uses the value of “id” column of the db table “rooms”</p>
M30F9	<p>In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M30F8 on Delete Request.</p>	N/A
M30F10	<p>After receiving delete request, a particular room will be retrieved from the database and will be removed. After then user will redirect to M30F4.</p>	<p>Find id, room title, description, room price, room availability from db table “rooms” by “id”</p> <p>Delete from db table “rooms” by “id”</p>

Booking Module(M31) – Service Provider (Hotel)		
Function	Description	Database
M31F1	<p>Booking module has two hyperlinks which will navigates to channels such as:</p> <ul style="list-style-type: none"> • Booking List: “/hotelDashboard/showhotelallboking” • Manage Booking: “/hotelDashboard/room-book-pending” 	N/A
M31F2	<p>In the M31F1, Booking List view page there will be a table having column Id, User Id, Room Id, Arrival Date, Departure Date. All the respective data of will be shown from database.</p> <p>Action column will be use to show action buttons such as Information and Delete.</p>	<p>List will use db table “roombooks” by req= “Approved” and columns id, user_id, room_id, fromdate, todote.</p>

M31F3	In the List of Module M31F2 an Information Button will be added to every row referring the route “/hotelDashboard/information{id}” . Here id parameter holds the user id of a row.	Uses the value of “user_id” column of the db table “roombooks”
M31F4	In the view there will be a table having column User Information and Room Information . User Information will show the Id, Name, Username, Gender, Email and Phone Number . Room Information will show the Room Id, Room Name, Room Price . All the respective data will be shown from database.	List will use db table “user” by user_id and columns id, username, firstname, lastname, gender, email, phone . List will use db table “rooms” by room_id and columns id, name, price .
M31F6	In the List of Module M31F2 a Delete Button will be added to every row referring the route “/hotelDashboard/bookingdelete{id}” . Here id parameter holds the id of a row.	Uses the value of “id” column of the db table “roombooks”
M31F7	In the view there'll be a confirmation. After confirm a virtual form will be submitted to the route of M31F6 on Delete Request.	N/A
M31F8	After receiving delete request, a particular booking list will be retrieved from the database and will be removed. After then user will redirect to M31F2 .	Find id, user_id, room_id, fromdate, todate from db table “roombooks” by “id” Delete from db table “roombooks” by “id” and make req = “Canceled”
M31F9	In the M31F1 , Manage Booking view page there will be a table having column Id, User Id, Room Id, Arrival Date, Departure Date . All the respective data will be shown from database. Action column will be use to show action buttons such as Accept and Decline .	List will use db table “roombooks” by req= “Pending” and columns id, user_id, room_id, fromdate, todate .
M31F10	In the List of Module M31F9 an Accept Button will be added to every row referring the route “/hotelDashboard/bookingapprove{id}” . Here id parameter holds the id of a row.	Uses the value of “id” column of the db table “rooms”
M31F11	In the view there'll be a confirmation. After confirm a virtual form will be submitted to the route of M31F10 on Accept Request	N/A
M31F12	After receiving accept request, a particular id will be accepted from the database and will be added. After then user will redirect to M31F9 .	Find id, user_id, room_id, fromdate, todate from db table “roombooks” by “id” Add to db table “roombooks” by “id” and make req = “Approved”

M31F13	In the List of Module M31F9 a Decline Button will be added to every row referring the route “/hotelDashboard/bookingdecline{id}” . Here id parameter holds the id of a row.	Uses the value of “id” column of the db table “roombooks”
M31F14	In the view there'll be a confirmation. After confirm a virtual form will be submitted to the route of M31F13 on Delete Request.	N/A
M31F15	After receiving delete request, a particular id will be retrieved from the database and will be removed. After then user will redirect to M31F9 .	Find id, user_id, room_id, fromdate, todate from db table “roombooks” by “id” Delete from db table “roombooks” by “id” and make req = “Decline”

Hotel Module (M32)		
Function	Description	Database
M32F1	A hotel list channel should be developed in the system in the route “/user/show_hotels” . There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be a html view page having Hotel ID, Hotel Name, Hotel Location, Hotel Address, Phone Number and Email . A hyperlink is added with 'Hotel Name' which will redirect to “/user/hotel_rooms{id}” .	Select all data from table “hotels” where req= “Approved” of database.
M32F2	Clicking the hyperlink would create a “GET” request to “/user/hotel_rooms{id}” with the id of that particular hotel. A room booking channel for that particular hotel should be developed in the system in the route “/user/hotel_rooms{id}” , where there'll be an html view page having Room ID, Hotel Id, Hotel Room Name, Hotel Room Price and Room Description . There will be two hyperlinks in the page. One for hotel facility channel which will redirect to “/user/hotel_facility” and another one is for booking hotels with room which will redirect to “/user/confirm_hotel{id}” .	Find hotel from database table “hotels” by “id” . Select all data from table “rooms” where availability= “Available” of database.
M32F3	Clicking the hyperlink named “Book” from M32F2 would create a “GET” request to “/user/confirm_hotel{id}” with the id of that particular hotel room. A hotel confirming channel should be developed in the system in the route “/user/confirm_hotel{id}” , where there'll be an html form having User ID, Room Id, Check-in, Check-out and a confirm button named which	Find hotel rooms from database table “rooms” by “id” . Store user_id= user_id, room_id= room_id, fromdate= fromdate, todate= todate from “rooms” table columns to

	will create a post request to “/user/confirm_hotel{id}”. And redirect to M32F1 .	respective columns of table “roombooks” where req= “Approved”
M32F4	Clicking the hyperlink named “Check Facility” from M32F2 should request to “/user/hotel_facility”. There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html table having columns ID, Facility Title, Description and Image . The table will show all the respective data from database.	Select all data from table “facilities” of database
M32F5	A hotel bookings channel should be developed in the system in the route “/user/hotel”. There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html table having columns SL No. and Information . The table will show all the respective data from database.	Select all data from table “roombooks” where req= “Approved” of database

Transport Management Module-Admin (M33)		
Function	Description	Database
M33F1	A pending transport list channel should be developed in the system in the route “/admin/transport-pending” where there’ll be an html table having columns ID, Name, Phone, Email, Type and Action . Contents of action column in each row will have two hyperlinks: <ul style="list-style-type: none"> Approve Referring the route “/admin/transport/approve{id}”. Decline Referring the route “/admin/transport/decline{id}”. 	Select all data from table “transports” of database Where req = “Pending” Uses the value of “id” column of the database table “transports”
M33F2	On clicking approve hyperlink beside each transport on M33F1 it would create a “GET” request to “/admin/transport/approve{id}” with the id of that particular transport An approve transport channel should be developed in the system in the route “/admin/transport/approve{id}” where there’ll be a html table having columns ID, Name, Phone, Email, Type and a confirm button which will create a post request to “/admin/transport/approve/{id}”. It would update the row of the transport of the table “transports” Column req = “Approved” And redirect to M33F1 with a get request.	Find transport from database table “transports” by “id” Updates req= “Approved”

M33F3	<p>On clicking decline hyperlink beside each transport on M33F1 it would create a “GET” request to “/admin/transport/decline{id}” with the id of that particular transport.</p> <p>A decline transport channel should be developed in the system in the route “/admin/transport/decline{id}” where there’ll be a html table having columns Id, Name, Phone, Email, Type and a confirm button which will create a post request to “/admin/transport/decline/{id}”. It would delete the row of the transport of the table “transports”</p> <p>And redirect to M33F1 with a get request.</p>	Delete a row from table “ transports ” by using id = “id”
M33F4	<p>A car company list channel should be developed in the system in the route “/admin/all-cars” where there’ll be a html table having columns Id, Name, Phone, Email, Type and Action.</p> <p>Contents of action column in each row will be referring the route “/admin/car/delete/{id}”.</p>	<p>Select data from table “transports” of database, where req = “Approved” and type = “Car”</p> <p>Uses the value of “id” column of the database table “transports”</p>
M33F5	<p>On clicking delete hyperlink beside each car company on M33F4 it would create a “GET” request to “/admin/car/delete/{id}” with the id of that particular car company.</p> <p>A car company account delete channel should be developed in the system in the route “/admin/car/delete/{id}” where there’ll be a html table having columns Id, Name, Phone, Email, Type and a confirm button which will create a post request to “/admin/car/delete/{id}”.</p> <p>It would delete the row of the car company of the table “transports”</p> <p>And redirect to M33F4 with a get request.</p>	Delete a row from table “ transports ” by using id = “id”
M33F6	<p>A flight company list channel should be developed in the system in the route “/admin/all-flight” where there’ll be a html table having columns Id, Name, Phone, Email, Type and Action.</p> <p>Contents of action column in each row will be referring the route “/admin/flight/delete/{id}”.</p>	<p>Select data from table “transports” of database, where req = “Approved” and type = “Flight”</p> <p>Uses the value of “id” column of the database table “transports”</p>
M33F7	<p>A car booking list channel should be developed in the system in the route “/admin/car-bookings” where there’ll be a html table having columns Id, User Id, Car Id, Departure Location, Arrival Location, Departure Date, Arrival Date Status and Action.</p> <p>Contents of action column in each row will be referring the route “/admin/car-bookings/{id}”.</p>	Select all data from table “ carbooks ” of database

M33F8	<p>A flight booking list channel should be developed in the system in the route “/admin/flight-bookings” where there’ll be a html table having columns Id, User Id, Air Id, Departure Location, Arrival Location, Departure Date, Arrival Date Status and Action. Contents of action column in each row will be referring the route “/admin/flight-bookings/{id}”.</p>	Select all data from table “airbooks” of database
--------------	---	---

Airplane Module(M35) – Service Provider (Flight)		
Function	Description	Database
M34F1	<p>Airplane module has two hyperlinks which will navigates to channels such as:</p> <ul style="list-style-type: none"> Add Airplane: “/flightDashboard/addflight” Manage Airplane: “/flightDashboard/manageflight” 	N/A
M34F2	<p>In the M34F1, Add Airplane view page there’ll be a html form having fields airplane name, airplane model, fare, type of airplane, airplane image and a add button to fire form submission to the same route with post request.</p>	N/A
M34F3	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> title: Required, length min. 2 model: Required, length min. 2 fare: Required, length min. 2 type: Required image: Required <p>If everything is fine title, model, fare, type, image will be stored in the database. After then user will redirect to M34F2.</p>	Store’s title = title, image = image, model = model, fare = fare, type= type to the db table “ airs ”
M34F4	<p>In the M34F1, Manage Airplane view page there will be a table having column S.no, Airplane Name, Airplane Model, Type of Airplane, Fare, Availability, Image. All the respective data will be shown from database. Action column will be use to show action buttons such as Edit and Delete.</p>	List will use db table “ airs ” by id and columns id, title, model, type, fare, image, availability .
M34F5	<p>In the List of Module M34F4 Edit Button will be added to every row referring the route “/flightDashboard/flightedit{id}”. Here id parameter holds the id of a row.</p>	Uses the value of “ id ” column of the db table “ airs ”

M34F6	In the view there'll be a html form having fields air title, air model and a update button to fire form submission to the same route with post request.	N/A
M34F7	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • title: Required, length min. 2 • model: Required, length min. 2 <p>If everything is fine title, price, model will be stored in the database. After then user will redirect to M34F4.</p>	<p>Find title, model from db table "airs" by "id"</p> <p>Store's title = title, model = model</p>
M34F8	In the List of Module M34F4 a Delete Button will be added to every row referring the route " /flightDashboard/flightdelete{id} ". Here id parameter holds the id of a row.	Uses the value of " id " column of the db table " airs "
M34F9	In the view there'll be a confirmation. After confirm a virtual form will be submitted to the route of M34F8 on Delete Request.	N/A
M34F10	<p>After receiving delete request, a particular air will be retrieved from the database and will be removed.</p> <p>After then user will redirect to M34F4.</p>	<p>Find id, title, model, room from db table "airs" by "id"</p> <p>Delete from db table "airs" by "id"</p>

Flight Type Module(M35) – Service Provider (Flight)		
Function	Description	Database
M35F1	In the M7F2 , Flight Type view page there will be a html form having fields Select Airplane, Select Type, Per Day Fare and a submit button to fire form submission to the same route with post request.	N/A
M35F2	<p>After receiving post request following form validation will be happen</p> <ul style="list-style-type: none"> • fare: Required, length min. 2 and max. 11 <p>If everything is fine airplane, type, fare will be stored in the database. After then user will redirect to M35F1.</p>	Store's title = title, type = type, fare = fare to the db table " airs "

Booking Module(M36) – Service Provider (Flight)		
Function	Description	Database
M36F1	<p>Booking module has two hyperlinks which will navigates to channels such as:</p> <ul style="list-style-type: none"> Booking List: “/flightDashboard/flight-booking-list” Manage Booking: “/flightDashboard/flight-pending-list” 	N/A
M36F2	<p>In the M36F1, Booking List view page there will be a table having column Id, User Id, Air Id, Arrival Location, Departure Location, Departure Date, Arrival Date. All the respective data of will be shown from database.</p> <p>Action column will be use to show action buttons such as Information and Delete.</p>	<p>List will use db table “airbooks” by req= “Approved” and columns id, user_id, air_id, fromloc, toloc, fromdate, todate.</p>
M36F3	<p>In the List of Module M36F2 an Information Button will be added to every row referring the route “/flightDashboard/information{id}”. Here id parameter holds the user id of a row.</p>	<p>Uses the value of “user_id” column of the db table “airbooks”</p>
M36F4	<p>In the view there will be a table having column User Information and Air Information. User Information will show the Id, Name, Username, Gender, Email and Phone Number. Air Information will show the Id, Title, Model, Type, Fare. All the respective data will be shown from database.</p>	<p>List will use db table “user” by user_id and columns id, username, firstname, lastname, gender, email, phone.</p> <p>List will use db table “airs” by room_id and columns id, title, model, type, fare.</p>
M36F6	<p>In the List of Module M36F2 a Delete Button will be added to every row referring the route “/flightDashboard/delete{id}”. Here id parameter holds the id of a row.</p>	<p>Uses the value of “id” column of the db table “airbooks”</p>
M36F7	<p>In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M36F6 on Delete Request.</p>	N/A
M36F8	<p>After receiving delete request, a particular booking list will be retrieved from the database and will be removed. After then user will redirect to M36F2.</p>	<p>Find id, user_id, air_id, fromdate, todate, fromloc, toloc from db table “airbooks” by “id”</p> <p>Delete from db table “airbooks” by “id” and make req = “Canceled”</p>

M36F9	In the M36F1 , Manage Booking view page there will be a table having column Id, User Id, Air Id, Arrival Location, Departure Location, Departure Date, Arrival Date . All the respective data will be shown from database. Action column will be use to show action buttons such as Accept and Decline .	List will use db table “airbooks” by req= “Pending” and columns id, user_id, air_id, fromloc, toloc, fromdate, todate .
M36F10	In the List of Module M36F9 an Accept Button will be added to every row referring the route “/flightDashboard/approve{id}” . Here id parameter holds the id of a row.	Uses the value of “id” column of the db table “airs”
M36F11	In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M36F10 on Accept Request	N/A
M36F12	After receiving accept request, a particular id will be accepted from the database and will be added. After then user will redirect to M36F9 .	Find id, user_id, air_id, fromdate, todate, fromloc, toloc from db table “airbooks” by “id” Add to db table “airbooks” by “id” and make req = “Approved”
M36F13	In the List of Module M36F9 a Decline Button will be added to every row referring the route “/flightDashboard/decline {id}” . Here id parameter holds the id of a row.	Uses the value of “id” column of the db table “airbooks”
M36F14	In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M36F13 on Delete Request.	N/A
M36F15	After receiving delete request, a particular id will be retrieved from the database and will be removed. After then user will redirect to M36F9 .	Find id, user_id, air_id, fromdate, todate, fromloc, toloc from db table “airbooks” by “id” Delete from db table “airbooks” by “id” and make req = “Decline”

Flight Availability Module(M37) – Service Provider (Flight)		
Function	Description	Database
M37F1	In the M7F2 , Flight Availability view page there will be a html form having fields Select Airplane, Availability and a submit button to fire form submission to the same route with post request.	N/A
M37F2	If everything is fine airplane, availability will be stored in the database. After then user will redirect to M37F1 .	Store's title = title, availability = availability , to the db table "airs"

Flight Module (M38)		
Function	Description	Database
M38F1	<p>A flight list channel should be developed in the system in the route "/user/flight_list".</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be a html table having columns ID, Title, Model, Type, Fare, Picture and Book. Under the "Book" column there will be a hyperlink which will redirect to "/user/flight/book{id}".</p>	Select all data from table "airs" where availability= "Available" of database.
M38F2	<p>Clicking the hyperlink under the "Book" column would create a "GET" request to "/user/flight/book{id}" with the id of that particular flight.</p> <p>A book flight channel should be developed in the system in the route "/user/flight/book{id}", where there'll be an html table having User ID, Flight Id, From, To, Check-in, Check-out and a confirm button named which will create a post request to "/user/flight/book{id}". And redirect to M38F1.</p>	<p>Find flight from database table "airs" by "id".</p> <p>Store user_id= user_id, air_id= air_id, fromloc= fromloc, toloc= toloc, fromdate= fromdate, todate= todate from "airs" table columns to respective columns of table "airbooks".</p>
M38F3	<p>A flight bookings channel should be developed in the system in the route "/user/flight".</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be a html table having columns SL No and Flight Information. The table will show all the respective data from database.</p>	Select all data from table "airbooks" where req= "Approved" of database

Car Module(M39) – Service Provider (Car)		
Function	Description	Database
M39F1	<p>Car module has two hyperlinks which will navigates to channels such as:</p> <ul style="list-style-type: none"> Add Car: “/carDashboard/addflight” Manage Car: “/carDashboard/manageflight” 	N/A
M39F2	In the M39F1 , Add Car view page there'll be a html form having fields car title, car model, driver name, per day fare, type of car, car image and a add button to fire form submission to the same route with post request.	N/A
M39F3	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> title: Required, length min. 2 model: Required, length min. 2 driver: Required, length min. 2 fare: Required, length min. 2 type: Required image: Required <p>If everything is fine title, model, fare, driver, type, image will be stored in the database. After then user will redirect to M39F2.</p>	Store's title = title, image = image, model = model, fare = fare, type= type, driver=driver to the db table “cars”
M39F4	<p>In the M39F1, Manage Car view page there will be a table having column S.no, car title, car model, driver name, per day fare, type of car, car image, availability. All the respective data will be shown from database.</p> <p>Action column will be use to show action buttons such as Edit and Delete.</p>	List will use db table “cars” by id and columns id, title, driver, model, type, fare, image, availability .
M39F5	In the List of Module M39F4 an Edit Button will be added to every row referring the route “/carDashboard/caredit {id}”. Here id parameter holds the id of a row.	Uses the value of “ id ” column of the db table “cars”
M39F6	In the view there'll be a html form having fields car title, car model, driver name and a update button to fire form submission to the same route with post request.	N/A

M39F7	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • title: Required, length min. 2 • model: Required, length min. 2 • driver: Required, length min. 2 <p>If everything is fine title, model, driver will be stored in the database. After then user will redirect to M39F4.</p>	<p>Find title, model, driver from db table “cars” by “id”</p> <p>Store’s title = title, model = model, driver=driver</p>
M39F8	<p>In the List of Module M39F4 a Delete Button will be added to every row referring the route “/carDashboard/cardelete{id}”. Here id parameter holds the id of a row.</p>	<p>Uses the value of “id” column of the db table “cars”</p>
M39F9	<p>In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M39F8 on Delete Request.</p>	N/A
M39F10	<p>After receiving delete request, a particular car will be retrieved from the database and will be removed. After then user will redirect to M39F4.</p>	<p>Find id, title, model, driver from db table “cars” by “id”</p> <p>Delete from db table “cars” by “id”</p>

Car Type Module(M40) – Service Provider (Car)		
Function	Description	Database
M40F1	<p>In the M8F2, Car Type view page there will be a html form having fields Select Car, Select Type, Per Day Fare and a submit button to fire form submission to the same route with post request.</p>	N/A
M40F2	<p>After receiving post request following form validation will be happen</p> <ul style="list-style-type: none"> • fare: Required, length min. 2 and max. 11 <p>If everything is fine title, type, fare will be stored in the database. After then user will redirect to M40F1.</p>	<p>Store’s title = title, type = type, fare = fare to the db table “cars”</p>

Booking Module(M41) – Service Provider (Car)		
Function	Description	Database
M41F1	<p>Booking module has two hyperlinks which will navigates to channels such as:</p> <ul style="list-style-type: none"> Booking List: “/carDashboard/car-booking-list” Manage Booking: “/carDashboard/car-pending-list” 	N/A
M41F2	<p>In the M41F1, Booking List view page there will be a table having column Id, User Id, Car Id, Arrival Location, Departure Location, Departure Date, Arrival Date. All the respective data of will be shown from database.</p> <p>Action column will be use to show action buttons such as Information and Delete.</p>	<p>List will use db table “carbooks” by req= “Approved” and columns id, user_id, car_id, fromloc, toloc, fromdate, todate.</p>
M41F3	<p>In the List of Module M41F2 an Information Button will be added to every row referring the route “/carDashboard/information{id}”. Here id parameter holds the user id of a row.</p>	<p>Uses the value of “user_id” column of the db table “carbooks”</p>
M41F4	<p>In the view there will be a table having column User Information and Car Information. User Information will show the Id, Name, Username, Gender, Email and Phone Number. Car Information will show the Id, Title, Model, Type, Fare, Driver. All the respective data will be shown from database.</p>	<p>List will use db table “user” by user_id and columns id, username, firstname, lastname, gender, email, phone.</p> <p>List will use db table “cars” by car_id and columns id, title, model, type, fare, driver.</p>
M41F6	<p>In the List of Module M41F2 a Delete Button will be added to every row referring the route “/carDashboard/delete{id}”. Here id parameter holds the id of a row.</p>	<p>Uses the value of “id” column of the db table “carbooks”</p>
M41F7	<p>In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M41F6 on Delete Request.</p>	N/A
M41F8	<p>After receiving delete request, a particular booking list will be retrieved from the database and will be removed. After then user will redirect to M41F2.</p>	<p>Find id, user_id, car_id, fromdate, todate, fromloc, toloc from db table “carbooks” by “id”</p> <p>Delete from db table “carbooks” by “id” and make req = “Canceled”</p>

M41F9	In the M41F1 , Manage Booking view page there will be a table having column Id, User Id, Car Id, Arrival Location, Departure Location, Departure Date, Arrival Date . All the respective data will be shown from database. Action column will be use to show action buttons such as Accept and Decline .	List will use db table “ carbooks ” by req= “ Pending ” and columns id, user_id, car_id, fromloc, toloc, fromdate, todate .
M41F10	In the List of Module M41F9 an Accept Button will be added to every row referring the route “ /carDashboard/approve{id} ”. Here id parameter holds the id of a row.	Uses the value of “ id ” column of the db table “ cars ”
M41F11	In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M41F10 on Accept Request	N/A
M41F12	After receiving accept request, a particular id will be accepted from the database and will be added. After then user will redirect to M41F9 .	Find id, user_id, car_id, fromdate, todate, fromloc, toloc from db table “ carbooks ” by “ id ” Add to db table “ carbooks ” by “ id ” and make req = “ Approved ”
M41F13	In the List of Module M41F9 a Decline Button will be added to every row referring the route “ /carDashboard/decline{id} ”. Here id parameter holds the id of a row.	Uses the value of “ id ” column of the db table “ carbooks ”
M41F14	In the view there’ll be a confirmation. After confirm a virtual form will be submitted to the route of M41F13 on Delete Request.	N/A
M41F15	After receiving delete request, a particular id will be retrieved from the database and will be removed. After then user will redirect to M41F9 .	Find id, user_id, car_id, fromdate, todate, fromloc, toloc from db table “ carbooks ” by “ id ” Delete from db table “ carbooks ” by “ id ” and make req = “ Decline ”

Car Availability Module(M42) – Service Provider (Car)		
Function	Description	Database
M42F1	In the M8F2 , Car Availability view page there will be a html form having fields Select Car, Availability and a submit button to fire form submission to the same route with post request.	N/A
M42F2	If everything is fine title, availability will be stored in the database. After then user will redirect to M42F1 .	Store's title = title, availability = availability , to the db table "cars"

Car Module (M43)		
Function	Description	Database
M43F1	<p>A car list channel should be developed in the system in the route "/user/Car_list".</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be a html table having columns ID, Car Title, Car Model, Driver Name, Type of Car, Fare, Picture and Book. Under the "Book" column there will be a hyperlink which will redirect to "/user/car/book{id}".</p>	Select all data from table "cars" where availability= "Available" of database.
M43F2	<p>Clicking the hyperlink under the "Book" column would create a "GET" request to "/user/car/book{id}" with the id of that particular car.</p> <p>A book car channel should be developed in the system in the route "/user/car/book{id}", where there'll be an html table having User ID, Car Id, From, To, Check-in, Check-out and a confirm button named which will create a post request to "/user/car/book{id}". And redirect to M43F1.</p>	<p>Find car from database table "cars" by "id".</p> <p>Store user_id= user_id, car_id= car_id, fromloc= fromloc, toloc= toloc, fromdate= fromdate, todate= todate from "cars" table columns to respective columns of table "carbooks".</p>
M43F3	<p>A car bookings channel should be developed in the system in the route "/user/car".</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be a html table having columns SL No. and Information. The table will show all the respective data from database.</p>	Select all data from table "carbooks" where req= "Approved" of database

Package Management Module-Admin (M44)		
Function	Description	Database
M44F1	<p>A pending package list channel should be developed in the system in the route “/admin/ package-pending” where there’ll be an html table having columns Id, Place, Location, Image, Description, Duration, Transport, Hotel, Cost, Status and Action. Contents of action column in each row will have two hyperlinks:</p> <ul style="list-style-type: none"> • Approve Referring the route “/admin/package/approve{id}”. • Decline Referring the route “/admin/ package /decline{id}”. 	<p>Select all data from table “packages” of database’ Where req = “Pending”</p> <p>Uses the value of “id” column of the database table “packages”</p>
M44F2	<p>On clicking approve hyperlink beside each package on M44F1 it would create a “GET” request to “/admin/package/approve{id}” with the id of that particular package.</p> <p>An approve package channel should be developed in the system in the route “/admin/package/approve{id}” where there’ll be a html table having Id, Place, Location, Description, Duration, Transport, Hotel, Cost, Status and a confirm button which will create a post request to “/admin/package/approve/{id}”.</p> <p>It would update the row of the package of the table “packages” Column req = “Approved” And redirect to M44F1 with a get request.</p>	<p>Find package from database table “packages” by “id”</p> <p>Updates req= “Approved”</p>
M44F3	<p>On clicking decline hyperlink beside each package on M44F1 it would create a “GET” request to “/admin/package/decline{id}” with the id of that particular package.</p> <p>A decline package channel should be developed in the system in the route “/admin/package/decline{id}” where there’ll be a html table having columns Id, Place, Location, Description, Duration, Transport, Hotel, Cost, Status and a confirm button which will create a post request to “/admin/package/decline/{id}”.</p> <p>It would delete the row of the package of the table “packages” And redirect to M44F1 with a get request.</p>	<p>Delete a row from table “packages” by using id = “id”</p>
M44F4	<p>A package channel should be developed in the system in the route “/admin/all-packages” where there’ll be a html table having columns Id, Place, Location, Description, Duration, Transport, Hotel, Cost, Status and Action. Contents of action column in each row will be referring the route “/admin/package/delete/{id}”.</p>	<p>Select data from table “packages” of database, where req = “Approved”.</p> <p>Uses the value of “id” column of the database table “packages”</p>

M44F5	<p>On clicking delete hyperlink beside each package on M44F4 it would create a “GET” request to “/admin/package/delete/{id}” with the id of that particular package.</p> <p>A package delete channel should be developed in the system in the route “/admin/package/delete/{id}” where there’ll be a html table having columns Id, Place, Location, Description, Duration, Transport, Hotel, Cost, Status and a confirm button which will create a post request to “/admin/package/delete/{id}”.</p> <p>It would delete the row of the package of the table “packages”</p> <p>And redirect to M44F4 with a get request.</p>	Delete a row from table “ packages ” by using id = “id”
M44F6	A package booking list channel should be developed in the system in the route “/admin/package-bookings” where there’ll be a html table having columns Id, User Id, Package Id, Place, Image and Status .	Select all data from table “packagebooks” of database
M44F7	<p>A package status channel should be developed in the system in the route “admin/package-status” where there’ll be a html form having Select Place, Status and a submit button to fire form submission to the same route with post request. Place would be selected from a drop down which would come from the “places” table in database. Status would be selected from a drop down which would contain 2 options:</p> <ol style="list-style-type: none"> 1. Upcoming 2. Completed 	Select data from table “ packages ” of database, where req = “Approved”
M44F8	Package will be updated (If submit button clicked) to the database using a post request to the same page with an update completion message.	<p>Find package from database table “packages”, where place = Select Place</p> <p>Updates status = Status</p>

Package Module - Employee (M45)		
Function	Description	Database
M45F1	<p>In the view the modules M5F1 and M5F2 will be added and there will be a form with a “Submit for Approval” button which will refer to the route “/employee/packageAdd”</p> <p>In the view a html form will contain “Place”, “location”, “image”, “description”, “duration”, “Include transport”, “Include hotel”, “cost” options.</p>	N/A

M45F2	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> • Places and location length must be less than 50 characters. • Image will be file type but stores as string in database. • Description length must be less than 1000 characters. • Duration length must be less than 100 characters. • Transport and Hotel are drag down field which contains yes/no options. • Cost length must be less than 5 characters and it should be decimal value. 	<p>Stores “place” = “place”, “location” = “location”, “image” = “image”, “description” = “description”, “duration” = “duration”, “transport” = “transport”, “hotel” = “hotel”, “cost” = “cost” in the packages table of database.</p> <p>Default value of “req” = “Pending”</p>
M45F3	<p>In the view of this modules of M5F1 and M5F2 will be added.</p> <p>A Manage Package channel should be developed in the system in the route “/employee/package” where there’ll be a html table having columns Id, Place, Location, Image, Description, Duration, Transport, Hotel, Cost, Status and Action. Contents of action column in each row will be referring the route “/employee/packageEdit{id}”. Here id parameter holds the package id of a row.</p> <p>Another action holds id of that package and deletes the row.</p>	<p>For delete it searches id=“id” and delete the row.</p> <p>For Edit it holds id and select the rows of that id.</p>
M45F4	<p>After clicking Edit button beside each row of M49F1 it would create a get request to “/employee/packageEdit{id}” with the id of that Package.</p> <p>A Package Edit channel will be developed in the route “employee/packageEdit{id}” where there will be a html form containing columns Id, Place, Location, Image, Description, Duration, Transport, Hotel, Cost, Status and a update button which will send a post request to “/employee/packageEdit{id}” that would update the row.</p>	<p>Select the id =“id” from database table packages and shows the rows for that id.</p>
M45F5	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> • Places and location length must be less than 50 characters. • Image will be file type but stores as string in database. • Description length must be less than 1000 characters. • Duration length must be less than 100 characters. • Transport, Hotels and Status field are required. <p>Cost length must be less than 5 characters and it should be decimal value.</p>	<p>Updates the database table packages by storing “place” = “place”, “location” = “location”, “image” = “image”, “description” = “description”, “duration” = “duration”, “transport” = “transport”, “hotel” = “hotel”, “cost” = “cost”.</p>

Package Module (M46)		
Function	Description	Database
M46F1	<p>A package channel should be developed in the system in the route “/user/package”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html table having columns ID, Place, Location, Image, Description, Duration, Transport, Hotel, Cost and Book. Under the “Book” column there will be a hyperlink which will redirect to “/user/package/book{id}”.</p>	Select all data from table “packages” where value of status= “Upcoming” of database.
M46F2	<p>Clicking the hyperlink under the “Book” column would create a “GET” request to “/user/package/book{id}” with the id of that particular package.</p> <p>A book package channel should be developed in the system in the route “/user/package/book{id}”, where there’ll be an html table having Package Id, Place, Location, Description, Duration, Transport, Hotel, Cost and a confirm button named “Book” which will create a post request to “/user/package/book{id}”. And redirect to M46F1.</p>	<p>Find package from database table “packages” by “id”.</p> <p>Store user_id = user_id, package_id = package_id, place= place, image= image where status= “Upcoming” from “packages” table columns to respective columns of table “packagebooks”.</p>
M46F3	<p>A travel history channel should be developed in the system in the route “/user/travel_history”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html table having columns ID, User ID, Package ID, Places and Image. The table will show all the respective data from database.</p>	Select all data from table “packagebooks” where of database

Advertisement Module - Employee (M47)		
Function	Description	Database
M47F1	<p>In the view the module M5F1 and M5F2 will be added with a post button.</p> <p>A html form will contain a column “upload ads file”</p>	Stores the file to database table “ ads ”
M47F2	<p>A table will be shown containing columns “ads”, “action” Contents of Action column refers to “/employee/adsdelete{id}”</p>	<p>Select all the rows from database table “ads”</p> <p>Uses the value of id column of the database table “ads”</p>
M47F3	<p>On clicking delete button beside each Ads row on E15F2 a GET request would be created on “/employee/adsdelete{id}” with the id of that particular ads.</p> <p>Ads delete channel would be developed in the system on</p>	Delete a row from “ ads ” by using “ id ” = “ id ”.

	<p>route “/employee/adsdelete{id}” where there will be a html table with columns “ads” and a Delete button which sends Post request on “/employee/adsdelete{id}”</p> <p>It will delete the Ads from “ads” table and redirect to M47F2 with a get request.</p>	
M47F4	<p>In the view the module M5F1 and M5F2 will be added with a post button.</p> <p>A html form will contain a column “Image”, “Message”</p>	Stores the file to database table “promos”
M47F5	<p>A table will be shown containing columns “image”, “message”, “action”</p> <p>Contents of Action column refers to “/employee/promodelete{id}”</p>	<p>Select all the rows from database table “promos”</p> <p>Uses the value of id column of the database table “promos”</p>
M47F6	<p>On clicking delete button beside each Promos row on M47F5 a GET request would be created on “/employee/promodelete{id}” with the id of that particular promos.</p> <p>Promo delete channel would be developed in the system on route “/employee/promodelete{id}” where there will be a html table with columns “image”, “message” and a Delete button which sends Post request on “/employee/promodelete{id}”</p> <p>It will delete the Promos from “promos” table and redirect to M47F5 with a get request.</p>	Delete a row from “promos” by using “id” = “id” .

Place Management Module-Admin (M48)		
Function	Description	Database
M48F1	<p>A pending place list channel should be developed in the system in the route “/admin/ place-pending” where there’ll be an html table having columns Id, Place, District, Image and Action.</p> <p>Contents of action column in each row will have two hyperlinks:</p> <ul style="list-style-type: none"> • Approve Referring the route “/admin/place/approve{id}”. • Decline Referring the route “/admin/place/decline{id}”. 	<p>Select all data from table “places” of database’ Where req = “Pending”</p> <p>Uses the value of “id” column of the database table “places”</p>
M48F2	<p>On clicking approve hyperlink beside each place on M48F1 it would create a “GET” request to “/admin/place/approve{id}” with the id of that particular place.</p> <p>An approve place channel should be developed in the system in the route “/admin/place/approve{id}”, where there’ll be an html table having Id, Place, District and a confirm button which will create a post request to “/admin/place/approve/{id}”.</p>	<p>Find place from database table “places” by “id”</p> <p>Updates req= “Approved”</p>

	It would update the row of the place of the table “place” Column req = “Approved” And redirect to M48F1 with a get request.	
M48F3	<p>On clicking decline hyperlink beside each place on M48F1 it would create a “GET” request to “/admin/place/decline{id}” with the id of that particular place.</p> <p>An decline place channel should be developed in the system in the route “/admin/place/decline{id}” where there’ll be a html table having columns Id, Place, District and a confirm button which will create a post request to “/admin/place/decline/{id}”.</p> <p>It would delete the row of the place of the table “places” And redirect to M48F1 with a get request.</p>	Delete a row from table “ places ” by using id = “id”
M48F4	<p>A place list channel should be developed in the system in the route “/admin/all-places” where there’ll be a html table having columns Id, Place, District, image and Action.</p> <p>Contents of action column in each row will be referring the route “/admin/place/delete/{id}”.</p>	<p>Select data from database table “place”, where req = “Approved”</p> <p>Uses the value of “id” column of the database table “places”</p>
M48F5	<p>On clicking delete hyperlink beside each place on M48F4 it would create a “GET” request to “/admin/place/delete/{id}” with the id of that particular place.</p> <p>A place delete channel should be developed in the system in the route “/admin/place/delete/{id}” where there’ll be a html table having columns Id, Place, District and a confirm button which will create a post request to “/admin/place/delete/{id}”.</p> <p>It would delete the row of the place of the table “places” And redirect to M48F4 with a get request.</p>	Delete a row from table “ places ” by using id = “id”

Place Module - Employee (M49)		
Function	Description	Database
M49F1	<p>In the view the modules M5F1 and M5F2 will be added and there will be a form with a “Submit for Approval” button which will refer to the route “/employee/placeAdd”</p> <p>In the view a html form will contain “Place”, “district”, “image” options.</p>	N/A
M49F2	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> Places and district length must be less than 50 characters. 	<p>Store “place” = “place”, “district” = “district”, “image” = “image” in the places table of database.</p> <p>Default value of “req” =</p>

	<ul style="list-style-type: none"> Image will be file type but stores as string in database. 	“Pending”
M49F3	<p>In the view of this modules of M5F1 and M5F2 will be added.</p> <p>A Manage Place channel should be developed in the system in the route “/employee/place” where there’ll be a html table having columns Id, Place, District, Image and Action. Contents of action column in each row will be referring the route “/employee/placeEdit{id}”. Here id parameter holds the place id of a row.</p> <p>Another action holds id of that place and deletes the row.</p>	<p>For delete it searches id=“id” and delete the row.</p> <p>For Edit it holds id and select the rows of that id.</p>
M49F4	<p>After clicking Edit button beside each row of M51F1 it would create a get request to “/employee/placeEdit{id}” with the id of that Place.</p> <p>A Place Edit channel will be developed in the route “employee/placeEdit{id}” where there will be a html form containing columns Id, Place, District, Image and a update button which will send a post request to “/employee/placeEdit{id}” that would update the row.</p>	Select the id =“id” from database table places and shows the rows for that id.
M49F5	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> Places and District length must be less than 50 characters. <p>Image will be file type but stores as string in database.</p>	Updates the database table places by storing “place” = “place”, “district” = “district”, “image” = “image.”

Accounts Module-Admin (M50)		
Function	Description	Database
M50F1	<p>A Income Statement channel should be developed in the system in the route “/admin/income-statement” where there’ll be a html table having columns Id, Month, Revenue, Cost and Profit.</p> <p>A report of Income Statement could be downloaded in pdf format with a button (download icon)</p>	Select all data from database table “incomes” .
M50F2	<p>A Transaction History channel should be developed in the system in the route “/admin/ transaction-history” where there’ll be a html table having columns Id, Date, Receiver, Username, Description and amount.</p> <p>A report of Income Statement could be downloaded in pdf format with a button (download icon)</p>	Select all data from database table “transactions” .
M50F3	<p>A Pending Salary and Festival Bonus channel should be developed in the system in the route “/admin/pending-salary”, where there’ll be an html table having columns Id, Employee Id, Username, Salary, Festival Bonus and Action.</p> <p>Contents of action column in each row will have two</p>	Select all data from database table “salaries” , where req = “Pending”

	<p>hyperlinks:</p> <ul style="list-style-type: none"> • Approve Referring the route “/admin/salary/approve{id}”. • Decline Referring the route “/admin/salary/decline{id}”. 	Uses the value of “id” column of the database table “salaries”
M50F4	<p>On clicking approve hyperlink beside each Salry and Festival Bonus on M50F3 it would create a “GET” request to “/admin/salary/approve{id}” with the id of that particular Salry and Festival Bonus.</p> <p>An approve Salry and Festival Bonus channel should be developed in the system in the route “/admin/salary/approve{id}”, where there’ll be an html table having Id, Employee Id, Username, Salary, Festival Bonus and a confirm button which will create a post request to “/admin/salary/approve/{id}”. It would update the row of the Salry and Festival Bonus of the table “salaries” Column req = “Approved” And redirect to M50F3 with a get request.</p>	<p>Find Salry and Festival Bonus from database table “salaries” by “id”</p> <p>Updates req= “Approved”</p>
M50F5	<p>On clicking decline hyperlink beside each Salry and Festival Bonus on M50F3 it would create a “GET” request to “/admin/salary/decline{id}” with the id of that particular place.</p> <p>An decline Salry and Festival Bonus channel should be developed in the system in the route “/admin/salary/decline{id}” where there’ll be a html table having Id, Employee Id, Username, Salary, Festival Bonus and a confirm button which will create a post request to “/admin/salary/decline/{id}”. It would delete the row of the Salry and Festival Bonus of the table “salaries” And redirect to M50F3 with a get request.</p>	<p>Delete a row from table “salaries” by using id = “id”</p>

Salary Module - Employee (M51)		
Function	Description	Database
M51F1	<p>In the view the modules M5F1 and M5F2 will be added and there will be a form with a “Send” button which will refer to the route “/employee/salary”</p> <p>In the view a html form will contain “Employee Id”, “Username”, “Salary”, “Bonus Percentage” options.</p>	N/A

M51F2	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> Employee Id length must be less than 5 characters. Employee Id must be Integer. Username length must be 3 to 20 characters. Salary and Bonus field must be decimal. Bonus percentage will calculate automatically: $\text{salary} * (\text{bonus percentage}) / 100$ and store in DB. 	<p>Store “Employee Id” = “employee_id”, “Username” = “username”, “Salary” = “salary”, “Bonus Percentage” = “bonus” in the salaries table of database.</p> <p>Default value of “req” = “Pending”</p>
M51F3	<p>In the view the module M5F1 and M5F2 will be added.</p> <p>A table will be shown containing the columns “Employee Id”, “Salary”, “Festival Bonus”.</p>	<p>Select the rows from database “salaries” table where “req” = “Approved”</p>
M51F4	<p>In the view a “Download Salary List” button will be added which redirects to “/employee/salary/download” and generates a salary report in pdf format.</p>	<p>Prints all the rows of database table “salaries” where “req” = “Approved”</p>

Income Statement Module - Employee (M52)		
Function	Description	Database
M52F1	<p>In the view the modules M5F1 and M5F2 will be added and there will be a form with a “Submit” button which will refer to the route “/employee/statementAdd”</p> <p>In the view a html form will contain “Month”, “Revenue”, “Cost” options.</p>	N/A
M52F2	<p>A validation will be happened for the form where all the columns are required. Validation includes:</p> <ul style="list-style-type: none"> Month will be month type and must be selected from a calendar option Revenue and Cost length must be less that 5 and it should be decimal value. 	<p>Store “Month” = “month”, “Revenue” = “revenue”, “Cost” = “cost” in the incomes table of database.</p>
M52F3	<p>In the view the module M5F1 and M5F2 will be added.</p> <p>A table will be shown containing the columns “Month”, “Revenue”, “Cost”, “Profit”.</p>	<p>Select the rows from database “incomes” table</p> <p>“profit” will be auto calculated from “revenue”-“cost”.</p>

M52F4	In the view a “Download Income Statement” button will be added which redirects to “/employee/statement/download” and generates an Income Statement report in pdf format.	Prints all the rows of database table “incomes”
--------------	--	--

Transaction History Module - Employee (M53)		
Function	Description	Database
M53F1	In the view the modules M5F1 and M5F2 will be added and there will be a form with a “Submit” button which will refer to the route “/employee/transactionAdd” In the view a html form will contain “Receiver Id”, “Receiver”, “Username”, “Description”, “Amount” options.	N/A
M53F2	A validation will be happened for the form where all the columns are required. Validation includes: <ul style="list-style-type: none"> Receiver and Username length must be 3 to 20 characters. Description must be less than 50 characters. Amounts must be decimal and character must be less than 5. Date will be current timestamp and cannot be manually input. 	Store “Receiver Id” = “receiver_id”, “Receiver” = “receiver”, “Username” = “username”, “Description” = “description”, “Amount”= “amount” in the transactions table of database.
M53F3	In the view the module M5F1 and M5F2 will be added. A table will be shown containing the columns “Date”, “Receiver Id”, “Receiver”, “Username”, “Description”, “Amount”	Select the rows from database “transactions” table.
M53F4	In the view a “Download Transaction History” button will be added which redirects to “/employee/transaction/download” and generates a Transaction History report in pdf format.	Prints all the rows of database table “transactions”

Transaction Module(M54) – Service Provider (Hotel)		
Function	Description	Database
M54F1	In the M6F2 , Transaction History view page there will be a table having column <u>Id</u>, Date, Receiver Id, Receiver, Username, Description, Amount . All the respective data will be shown from database. A report of Transaction Report could be downloaded in pdf format with a button (download icon)	List will use db table “transactions” by receiver_id and receiver and columns id, date, username, description, amount .

Transaction Module(M55) – Service Provider (Flight)		
Function	Description	Database
M55F1	<p>In the M7F2, Transaction History view page there will be a table having column Id, Date, Receiver Id, Receiver, Username, Description, Amount. All the respective data will be shown from database.</p> <p>A report of Transaction Report could be downloaded in pdf format with a button (download icon)</p>	List will use db table “ transactions ” by receiver_id and receiver and columns id, date, username, description, amount .

Transaction Module(M56) – Service Provider (Car)		
Function	Description	Database
M56F1	<p>In the M8F2, Transaction History view page there will be a table having column Id, Date, Receiver Id, Receiver, Username, Description, Amount. All the respective data will be shown from database.</p> <p>A report of Transaction Report could be downloaded in pdf format with a button (download icon)</p>	List will use db table “ transactions ” by receiver_id and receiver and columns id, date, username, description, amount .

Support Module-Admin (M57)		
Function	Description	Database
M57F1	<p>A Support channel should be developed in the system in the route “/admin/support” where there’ll be an html table having columns Id, Name, Contact No., Email, Message and Action. Contents of action column in each row will be referring the route “/admin/support/delete/{id}”.</p>	Select all data from database table “ supports ”.
M57F2	<p>On clicking delete hyperlink beside each support on M57F1 it would create a “GET” request to “/admin/support/delete/{id}” with the id of that particular support.</p> <p>A support delete channel should be developed in the system in the route “/admin/support/delete/{id}” where there’ll be an html table having columns Id, Name, Contact No., Email, Message and a confirm button which will create a post request to “/admin/support/delete/{id}”. It would delete the row of the support of the table “supports”, and redirect to M57F1 with a get request.</p>	Delete a row from table “ supports ” by using id = “id”

Support Module(M58) – Service Provider (Hotel)		
Function	Description	Database
M58F1	In the M6F2 , Support view page there will be a html form having fields hotel name, phone number, email, message and a submit button to fire form submission to the same route with post request.	N/A
M58F2	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • username: Required, length min. 2 and max. 30 • phone: Required, length min. 2 and max. 15 • email: Required • message: Required, length min. 2 <p>If everything is fine username, phone, email, message will be stored in the database. After then user will redirect to M58F1.</p>	Store's username = username, phone = phone, email = email, message = message to the db table "supports"

Support Module(M59) – Service Provider (Flight)		
Function	Description	Database
M59F1	In the M7F2 , Support view page there will be a html form having fields company name, phone number, email, message and a submit button to fire form submission to the same route with post request.	N/A
M59F2	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • username: Required, length min. 2 and max. 30 • phone: Required, length min. 2 and max. 15 • email: Required • message: Required, length min. 2 <p>If everything is fine username, phone, email, message will be stored in the database. After then user will redirect to M59F1.</p>	Store's username = username, phone = phone, email = email, message = message to the db table "supports"

Support Module(M60) – Service Provider (Car)		
Function	Description	Database
M60F1	In the M8F2 , Support view page there will be a html form having fields company name, phone number, email, message and a submit button to fire form submission to the same route with post request.	N/A
M60F2	<p>After receiving post request following form validation will be happen,</p> <ul style="list-style-type: none"> • username: Required, length min. 2 and max. 30 • phone: Required, length min. 2 and max. 15 • email: Required • message: Required, length min. 2 <p>If everything is fine username, phone, email, message will be stored in the database. After then user will redirect to M60F1.</p>	Store's username = username, phone = phone, email = email, message = message to the db table “supports”

Contact Us Module (M61)		
Function	Description	Database
M61F1	<p>A contact channel should be developed in the system in the route “/user/contact”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There'll be a html form having fields Name, Phone Number, Email and Message and a submit button named “Send Message” to fire form submission to the same route with post request.</p>	N/A
M61F2	<p>A backend form validation will occur which will validate the credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <p>fields Name, Phone Number, Email and Message field validation.</p> <p>For any validation error an error message will be shown. For successful validation the review will be added to the database.</p>	<p>Match database from table “supports”.</p> <p>Store username= username, phone = phone, com, email= email, message= message respective columns of table “supports”.</p>

Policy Module-Admin (M62)		
Function	Description	Database
M62F1	A Privacy Policy channel should be developed in the system in the route “/admin/policy” where there’ll be policy contents from database are written and an edit hyperlink which will be referring the route “/admin/edit-policy{id}”.	Select all data from database table “ policies ”.
M62F2	<p>On clicking edit hyperlink M62F1 it would create a “GET” request to “/admin/ edit-policy{id}”, with the id of that particular support.</p> <p>A Edit Privacy Policy channel should be developed in the system in the route “/admin/ edit-policy{id}” where there’ll be a html form having field Write Privacy Policies Here, which will be filled with data from the database and a submit button to fire form submission to the same route with post request.</p>	Select all data from database table “ policies ”, where id = id
M62F3	<p>A backend form validation will occur which will validate the credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> • Write Privacy Policies Here field cannot be empty. <p>For any validation error an error message will be shown. For successful validation the privacy policy will be updated and will be redirected to M62F1 with a get request.</p>	<p>Find privacy policy from database table “policies” by “id”</p> <p>Updates policy = policyForm</p>

Privacy Policy Module (M63)		
Function	Description	Database
M63F1	<p>A privacy policy channel should be developed in the system in the route “/user/privay_policy”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. In this view page all the respective data will be shown from database.</p>	Select all data from table “policies” of database.

Travel Guideline Module-Admin (M64)		
Function	Description	Database
M64F1	A Travel Guideline channel should be developed in the system in the route “/admin/guidelines” where there’ll be guidelines contents from database are written and an edit hyperlink which will be referring the route “/admin/edit-guidelines{id}”.	Select all data from database table “guidelines”.
M64F2	<p>On clicking edit hyperlink M64F1 it would create a “GET” request to “/admin/ edit-guidelines{id}”, with the id of that particular guideline.</p> <p>A Edit Travel Guidelines channel should be developed in the system in the route “/admin/ edit-guidelines{id}” where there’ll be a html form having field Write Travel Guideline here, which will be filled with data from the database and a submit button to fire form submission to the same route with post request.</p>	Select all data from database table “guidelines”, where id = id
M64F3	<p>A backend form validation will occur which will validate the credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> Write Travel Guidelines here field cannot be empty. <p>For any validation error an error message will be shown. For successful validation the Travel Guidelines will be updated and will be redirected to M64F1 with a get request.</p>	<p>Find privacy policy from database table “guidelines” by “id”</p> <p>Updates guideline = guidelineForm</p>

Travel Guideline Module (M65)		
Function	Description	Database
M65F1	<p>A travel guideline channel should be developed in the system in the route “/user/guideline”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. In this view page all the respective data will be shown from database.</p>	Select all data from table “guidelines” of database.

Notification Module (M66)		
Function	Description	Database
M66F1	<p>A notification channel should be developed in the system in the route “/user/notification”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. In this view page the contents will be shown in two different tables. One table will show Ads and the other one will show Promos.</p>	Select all data from table “ads” and “promos” of database.

About Us Module-Admin (M67)		
Function	Description	Database
M67F1	A About Us channel should be developed in the system in the route “/admin/about” where there’ll be guidelines contents from database are written and an edit hyperlink which will be referring the route “/admin/edit-about{id}”.	Select all data from database table “abouts”.
M67F2	<p>On clicking edit hyperlink M67F1 it would create a “GET” request to “/admin/ edit-about{id}”, with the id of that particular about us.</p> <p>An Edit Travel Guidelines channel should be developed in the system in the route “/admin/ edit-about{id}” where there’ll be a html form having field Write About Us page content here, which will be filled with data from the database and a submit button to fire form submission to the same route with post request.</p>	Select all data from database table “abouts”, where id = id
M67F3	<p>A backend form validation will occur which will validate the credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <ul style="list-style-type: none"> Write About Us page content here field cannot be empty. <p>For any validation error an error message will be shown. For successful validation the About Us page content will be updated and will be redirected to M67F1 with a get request.</p>	<p>Find privacy policy from database table “abouts” by “id”</p> <p>Updates about = aboutForm</p>

About Us Module (M68)		
Function	Description	Database
M68F1	<p>A about us channel should be developed in the system in the route “/user/about”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. In this view page all the respective data will be shown from database.</p>	Select all data from table “abouts” of database.

Review Module - Employee (M69)		
Function	Description	Database
M69F1	<p>In the view the module M5F1 and M5F2 will be added.</p> <p>A table will be added which contains column “Service Id”, “Service Type”, “Company Name”, “User Id”, “Ratings”, “Message”.</p>	Shows all the rows from database table “ reviews ”

Review Module(M70) – Service Provider (Hotel)		
Function	Description	Database
M70F1	<p>In the M6F2, Check Review view page there will be a table having column Id, Service Id, Service Type, Company, User Id, Rating, Message. All the respective data will be shown from database.</p>	List will use db table “ reviews ” by service_id and service_type and columns id, company, user_id, rating, message .

Review Module(M71) – Service Provider (Flight)		
Function	Description	Database
M71F1	<p>In the M7F2, Check Review view page there will be a table having column Id, Service Id, Service Type, Company, User Id, Rating, Message. All the respective data will be shown from database.</p>	List will use db table “ reviews ” by service_id and service_type and columns id, company, user_id, rating, message .

Review Module(M72) – Service Provider (Car)		
Function	Description	Database
M72F1	In the M8F2 , Check Review view page there will be a table having column Id, Service Id, Service Type, Company, User Id, Rating, Message . All the respective data will be shown from database.	List will use db table “ reviews ” by service_id and service_type and columns id, company, user_id, rating, message .

Review Module (M73)		
Function	Description	Database
M73F1	<p>A review channel should be developed in the system in the route “/user/give_review”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. There’ll be a html form having fields Service ID, Service Type, Company Name, User ID, Rating and Message and a submit button named “Give Review” to fire form submission to the same route with post request.</p>	N/A
M73F2	<p>A backend form validation will occur which will validate the credentials submitted if they are secure to pass to the database for validation. Validations includes:</p> <p>Service ID, Service Type, Company Name, User ID, Rating and Message field validation.</p> <p>For any validation error an error message will be shown. For successful validation the review will be added to the database.</p>	<p>Match database from table “reviews” and column “user_id”,</p> <p>Store service_id= service_id, service_type = service_type, company= company, user_id= user_id, rating= rating, msg= msg respective columns of table “reviews”.</p> <p>Use the value of “id” column of the database table “users”</p>

FAQ Module - Employee (M74)		
Function	Description	Database
M74F1	<p>In the view the module M5F1 and M5F2 will be added with a post button.</p> <p>A html form will contain a column “Questions”, “Answers”</p>	Stores the file to database table “faqs”
M74F2	<p>A table will be shown containing columns “questions”, “answers”, “action”</p> <p>Contents of Action column refers to “/employee/faqdelete{id}”</p>	<p>Select all the rows from database table “faqs”</p> <p>Uses the value of id column of the database table “faqs”</p>
M74F3	<p>On clicking delete button beside each FAQs row on E18F2 a GET request would be created on “/employee/faqdelete{id}” with the id of that particular faqs.</p> <p>FAQ delete channel would be developed in the system on route “/employee/faqdelete{id}” where there will be a html table with columns “question”, “answer” and a Delete button which sends Post request on “/employee/faqdelete{id}”</p> <p>It will delete the FAQs from “faqs” table and redirect to M74F2 with a get request.</p>	Delete a row from “faqs” by using “id” = “id”.

FAQ Module (M75)		
Function	Description	Database
M75F1	<p>An faq channel should be developed in the system in the route “/user/faq”.</p> <p>There should be a header from M9F1 which will contain the necessary hyperlinks and channels. In this view page all the respective data will be shown from database.</p>	Select all data from table “faqs” of database.

Access Control:

Role name	Module(M), Functions(F)
Admin	<ul style="list-style-type: none">• M1F1, M1F2, M1F3• M2F1• M4F1, M4F2, M4F3• M10F1• M15F1, M15F2• M21F1, M21F2, M21F3, M21F4• M28F1, M28F2, M28F3, M28F4, M28F5, M28F6• M33F1, M33F2, M33F3, M33F4, M33F5, M33F6, M33F7, M33F8, M33F9• M31F4• M36F4• M41F4• M44F1, M44F2, M44F3, M44F4, M44F5, M44F6, M44F7, M44F8• M48F1, M48F2, M48F3, M48F4, M48F5• M50F1, M50F2, M50F3, M50F4, M50F5• M57F1, M57F2• M62F1, M62F2, M62F3• M64F1, M64F2, M64F3• M67F1, M67F2, M67F3

Role name	Module(M), Functions(F)
Employee	<ul style="list-style-type: none">• M1F1, M1F2, M1F3• M2F1• M5F1, M5F2• M11F1• M16F1, M16F2, M16F3• M26F1, M26F2, M26F3, M26F4, M26F5, M26F6• M27F1, M27F2, M27F3, M27F4• M45F1, M45F2, M45F3, M45F4, M45F5• M47F1, M47F2, M47F3, M47F4, M47F5, M47F6• M49F1, M49F2, M49F3, M49F4, M49F5• M51F1, M51F2, M51F3, M51F4• M52F1, M52F2, M52F3, M52F4• M53F1, M53F2, M53F3, M53F4• M69F1• M74F1, M74F2, M74F3

Role name	Module(M), Functions(F)
Service Provider	<ul style="list-style-type: none"> • M1F1, M1F2, M1F3 • M2F1 • M3F1, M3F4, M3F5, M3F6, M3F7 • M6F1, M6F2 • M7F1, M7F2 • M8F1, M8F2 • M12F1, M12F2 • M13F1, M13F2 • M14F1, M14F2 • M17F1, M17F2 • M18F1, M18F2 • M19F1, M19F2 • M29F1, M29F2, M29F3, M29F4, M29F5, M29F6, M29F7, M29F8, M29F9, M29F10 • M30F1, M30F2, M30F3, M30F4, M30F5, M30F6, M30F7, M30F8, M30F9, M30F10 • M31F1, M31F2, M31F3, M31F4, M31F6, M31F7, M31F8, M31F9, M31F10, M31F11, M31F12, M31F13, M31F14, M31F15 • M34F1, M34F2, M34F3, M34F4, M34F5, M34F6, M34F7, M34F8, M34F9, M34F10 • M35F1, M35F2 • M36F1, M36F2, M36F3, M36F4, M36F6, M36F7, M36F8, M36F9, M36F10, M36F11, M36F12, M36F13, M36F14, M36F15 • M37F1, M37F2 • M39F1, M39F2, M39F3, M39F4, M39F5, M39F6, M39F7, M39F8, M39F9, M39F10 • M40F1, M40F2 • M41F1, M41F2, M41F3, M41F4, M41F6, M41F7, M41F8, M41F9, M41F10, M41F11, M41F12, M41F13, M41F14, M41F15 • M42F1, M42F2 • M54F1 • M55F1 • M56F1 • M58F1, M58F2 • M59F1, M59F2 • M60F1, M60F2 • M70F1 • M71F1 • M72F1

Role name	Module(M), Functions(F)
User	<ul style="list-style-type: none"> • M1F1, M1F2, M1F3 • M2F1 • M3F1, M3F2, M3F3 • M9F1 • M20F1, M20F2 • M22F1, M22F2, M22F3, M22F4 • M23F1, M23F2 • M24F1, M24F2, M23F3, M23F4 • M25F1, M25F2, M25F3 • M32F1, M32F2, M32F3, M32F4, M32F5 • M38F1, M38F2, M38F3 • M43F1, M43F2, M43F3 • M46F1, M46F2, M46F3 • M61F1, M61F2 • M63F1 • M65F1 • M66F1 • M68F1 • M73F1, M73F2 • M75F1