

Travel Management System

Capstone Project

Case Study

By

RASHMI R

INTRODUCTION

The **Travel Management System** is a web-based application designed to automate and simplify the process of managing travel packages, hotel bookings, and user reservations.

It provides a centralized platform where **users can browse travel packages, book trips, and track booking status**, while **administrators manage packages, hotels, and booking approvals**.

The system reduces manual work, improves accuracy, and ensures a smooth booking workflow between users and administrators.

This project is developed using **Spring Boot, MySQL, HTML, CSS, JavaScript, and REST APIs**.

ABSTRACT

The **Travel Management System (TMS)** is developed to manage travel-related activities such as package creation, hotel management, user bookings, and approval workflows.

The system allows users to **register, log in, view packages, book trips, and monitor booking status**, while administrators can **add hotels, create packages, approve or cancel bookings, and monitor system statistics**.

The main goal of this project is to demonstrate **real-world CRUD operations, role-based access control, and client-server interaction** using Spring Boot and MYSQL. The project follows a layered architecture consisting of **Controller, Service, DAO, and Database layers**.

CLIENT REQUIREMENTS

The client requires a Travel Management System with the following features:

- Admin should be able to:
 - Add, update, delete, and view travel packages
 - Add and manage hotels
 - View all bookings
 - Approve or cancel user bookings
 - View dashboard statistics (total users, bookings, hotels, packages)
- User should be able to:
 - Register and log in
 - View available travel packages
 - Book travel packages
 - View booking history and status
 - Cancel bookings (before approval)

TECHNICAL FEATURES

- Proper **Login and Registration**
- **Role-based authentication** (Admin / User)
- **REST APIs** for data communication
- CRUD operations on:
 - ✓ Users
 - ✓ Hotels
 - ✓ Travel Packages
 - ✓ Bookings
- Booking approval workflow

- Dashboard with real-time statistics
- Exception handling and validation
- Clean UI with HTML, CSS, and JavaScript

TECHNOLOGIES AND TOOLS USED

- ✓ **Programming Language:** Java (JDK 17)
- ✓ **Framework:** Spring Boot
- ✓ **Frontend:** HTML, CSS, JavaScript
- ✓ **Database:** MySQL 8.0
- ✓ **IDE:** Spring Tool Suite (STS)
- ✓ **Web Server:** Apache Tomcat (embedded)
- ✓ **Build Tool:** Maven
- ✓ **API Testing:** Postman
- ✓ **Browser:** Google Chrome

PROJECT MODULES

- 🚧 **Admin Module**
- 🚧 **User Module**
- 🚧 **Hotel Module**
- 🚧 **Travel Package Module**
- 🚧 **Booking Module**

MODULE DESCRIPTION

1. Admin Module

The **Admin Module** is responsible for managing and monitoring the entire Travel Management System. It provides administrative control over users, travel packages, hotels, and bookings.

Features:

- Admin login with authentication
- Dashboard overview showing:
 - Total users
 - Total hotels
 - Total travel packages
 - Total bookings
- Add new travel packages
- Add hotel details
- View all user bookings
- Approve or cancel user bookings
- Monitor pending bookings

Purpose:

This module ensures that all system activities are properly controlled and managed. Only authorized admin users can access this module.

2. User Module

The **User Module** allows customers to interact with the system, explore travel options, and make bookings.

Features:

- User registration
- User login and logout
- View available travel packages
- Book a travel package
- View personal booking history
- Check booking status (BOOKED / APPROVED / CANCELLED)
- Cancel booking (if booking is still pending)

Purpose:

This module provides a smooth interface for users to plan and manage their travel bookings.

3. Travel Package Module

The **Travel Package Module** manages all travel packages offered by the system.

Features:

- Add new travel packages (Admin only)
- View list of all travel packages
- Display package details:
 - Destination
 - Price
 - Number of days
 - Description
 - Associated hotel
- Delete travel packages (Admin only)

Purpose:

This module acts as the core of the application by defining travel plans that users can browse and book.

4. Hotel Module

The **Hotel Module** manages hotel details linked to travel packages.

Features:

- Add hotel details (Admin only)
- Store hotel information such as:
 - Hotel name
 - Location
 - Price per day
- Associate one hotel with a travel package
- Display hotel details in bookings

Purpose:

This module ensures that each travel package is associated with proper accommodation details, enhancing the realism of the system.

5. Booking Module

The **Booking Module** handles the entire booking workflow between users and administrators.

Features:

- User books a travel package
- Booking status initially set to **BOOKED**
- Admin reviews and updates booking status:

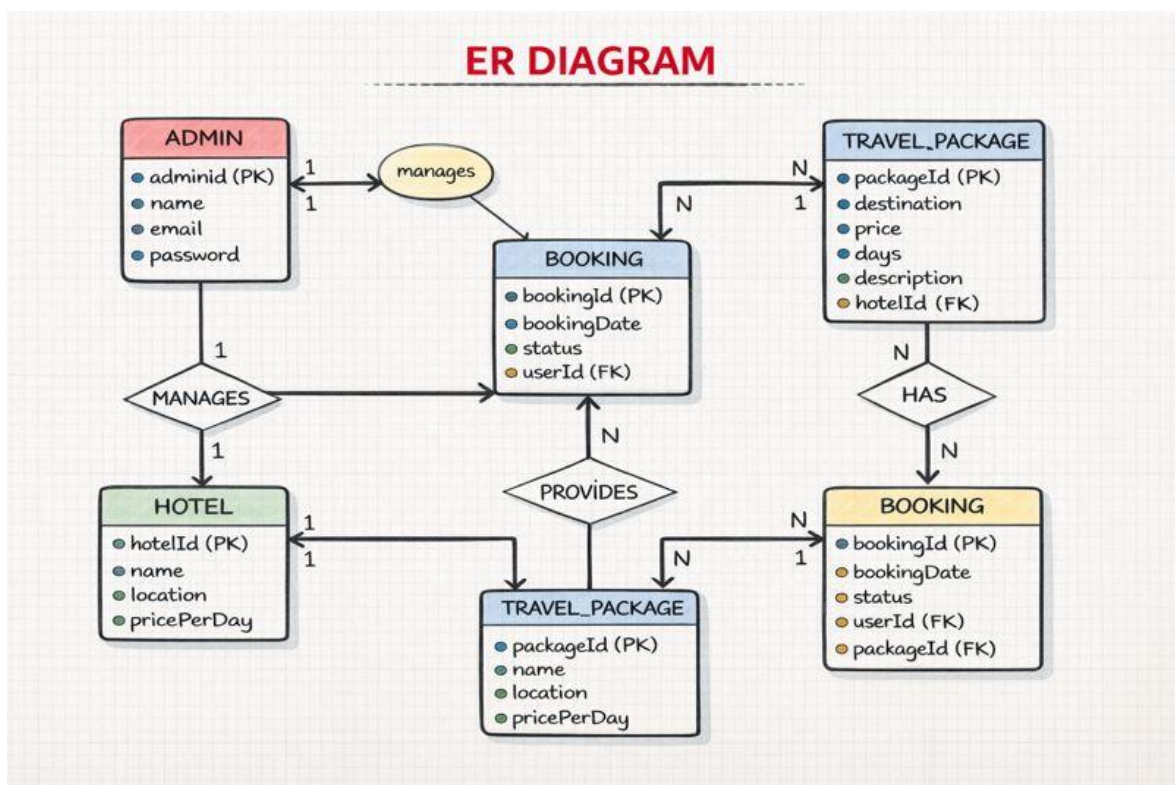
- APPROVED
- CANCELLED

- User views booking history and status
- Admin views all bookings and pending requests

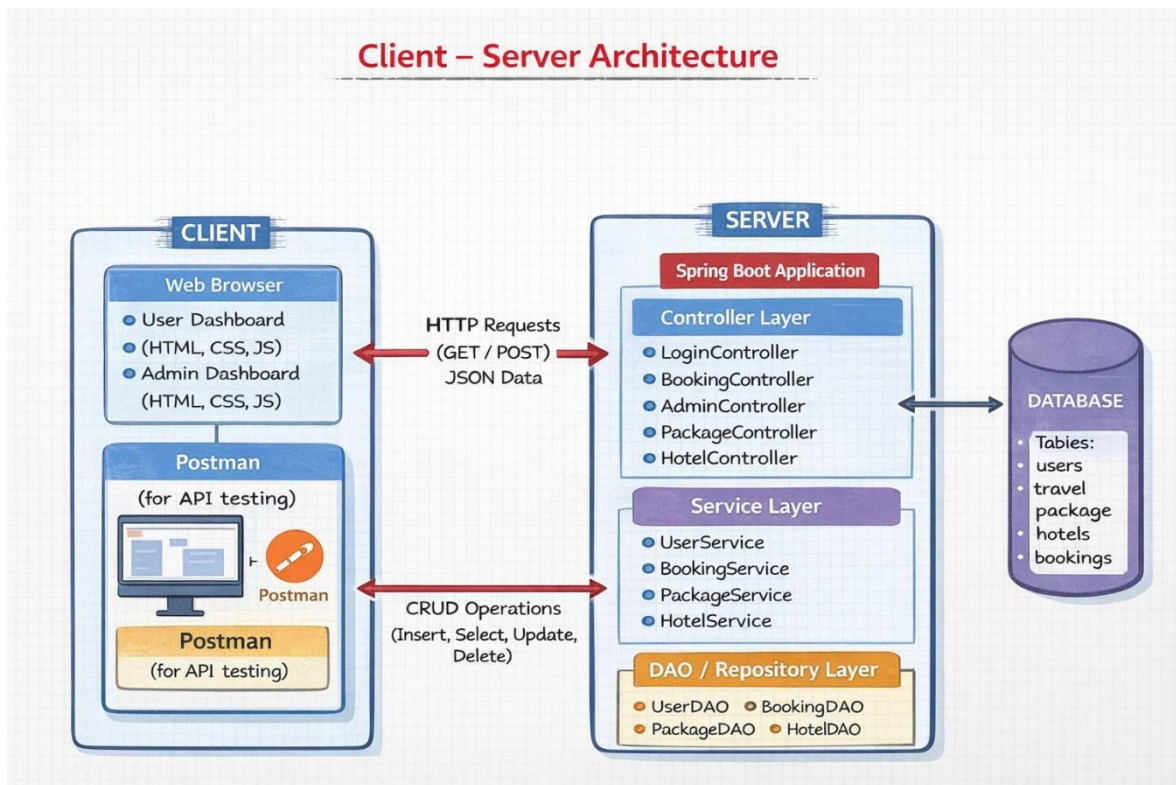
Purpose:

This module manages the interaction between users and admins, ensuring bookings are properly tracked and approved.

ER DIAGRAM



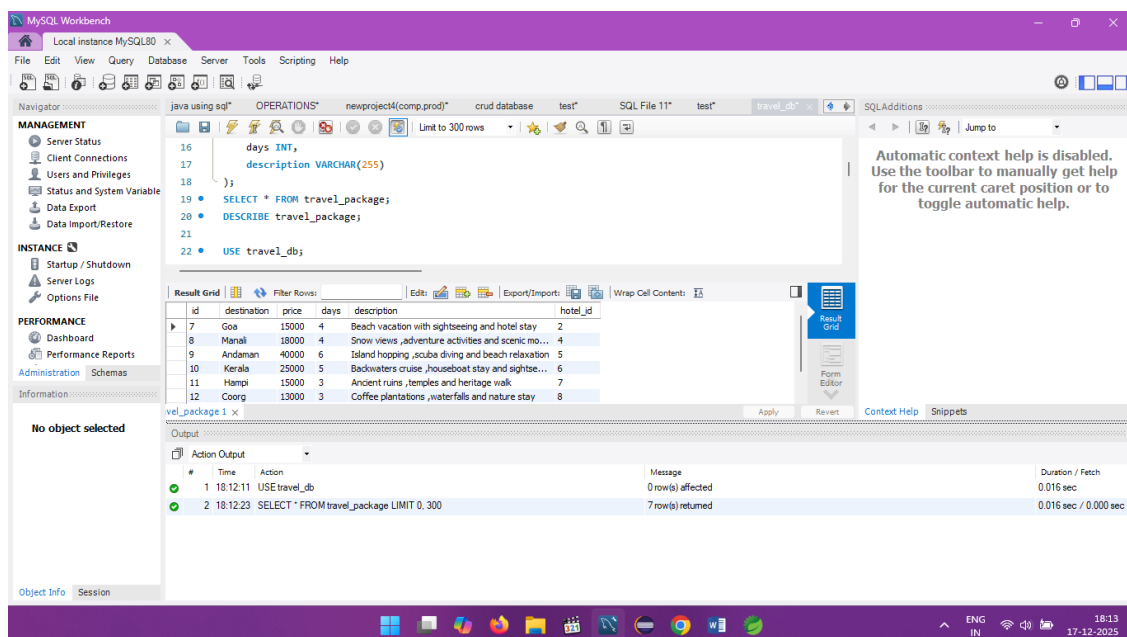
CLIENT-SERVER ARCHITECTURE



DATA DICTIONARY

TABLES OF SQL DATABASES

The Travel Management System uses MySQL as the backend database. The database consists of four main tables: Users, Hotels, Travel Packages, Bookings. These tables are connected using foreign key relationships to maintain data integrity.



Travel-Packages Table

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: java using sql* OPERATIONS* newproject4(comp.prod)* crud database test* SQL File 11* test* travel_db*

MANAGEMENT: Server Status, Client Connections, Users and Privileges, Status and System Variable, Data Export, Data Import/Restore

INSTANCE: Startup / Shutdown, Server Logs, Options File

PERFORMANCE: Dashboard, Performance Reports, Administration, Schemas

Information: No object selected

SQL Editor:

```

34 • INSERT INTO users (name, email, password)
35 • VALUES ('Admin', 'admin@gmail.com', 'admin123');
36
37 • SELECT * FROM users;
38
39 • USE travel_db;
40

```

Result Grid:

| id | name | email | password | role |
|----|-----------|-------------------|----------|-------|
| 1 | Admin | admin@gmail.com | admin123 | ADMIN |
| 5 | Test User | user@gmail.com | user123 | USER |
| 6 | Rashmi R | rashmi@gmail.com | 12345 | USER |
| 12 | Kashika | kashika@gmail.com | 12165 | USER |

users 2 x

Output:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|-------------------|-----------------------|
| 1 | 18:12:11 | USE travel_db | 0 row(s) affected | 0.016 sec |
| 2 | 18:12:23 | SELECT * FROM travel_package LIMIT 0, 300 | 7 row(s) returned | 0.016 sec / 0.000 sec |
| 3 | 18:14:56 | SELECT * FROM users LIMIT 0, 300 | 4 row(s) returned | 0.000 sec / 0.000 sec |

Object Info: Session

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Users Table

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: java using sql* OPERATIONS* newproject4(comp.prod)* crud database test* SQL File 11* test* travel_db*

MANAGEMENT: Server Status, Client Connections, Users and Privileges, Status and System Variable, Data Export, Data Import/Restore

INSTANCE: Startup / Shutdown, Server Logs, Options File

PERFORMANCE: Dashboard, Performance Reports, Administration, Schemas

Information: No object selected

SQL Editor:

```

56
57 • SELECT * FROM hotels;
58

```

Result Grid:

| id | name | location | price_per_day |
|----|---------------------|----------|---------------|
| 1 | Taj Residency | Goa | 4500 |
| 2 | Taj Residency | Goa | 3500 |
| 3 | Taj Residency | Goa | 3500 |
| 4 | Snow Peak Inn | Manali | 4000 |
| 5 | Coral Island Resort | Andaman | 5500 |

hotels 3 x

Output:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---|-----------------------|
| 1 | 18:12:11 | USE travel_db | 0 row(s) affected | 0.016 sec |
| 2 | 18:12:23 | SELECT * FROM travel_package LIMIT 0, 300 | 7 row(s) returned | 0.016 sec / 0.000 sec |
| 3 | 18:14:56 | SELECT * FROM users LIMIT 0, 300 | 4 row(s) returned | 0.000 sec / 0.000 sec |
| 4 | 18:15:29 | INSERT INTO users (name, email, password) VALUES ('Admin', 'admin@gmail.com', 'admin123') | Error Code: 1364. Field 'role' doesn't have a default value | 0.000 sec |
| 5 | 18:16:16 | SELECT * FROM hotels LIMIT 0, 300 | 9 row(s) returned | 0.000 sec / 0.000 sec |

Object Info: Session

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Hotels Table

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variable
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports

Administration Schemas

Information

No object selected

Object Info Session

25* java using sql* OPERATIONS* newproject4(comp.prod)* crud database test* SQL File 11* test* travel_db*

Limit to 300 rows

```

76 status VARCHAR(20) DEFAULT 'BOOKED',
77
78 FOREIGN KEY (package_id) REFERENCES travel_package(id)
79 );
80
81 SELECT * FROM bookings;
82

```

Result Grid

| | id | user_email | package_id | booking_date | status |
|---|----|----------------|------------|---------------------|----------|
| 1 | 1 | user@gmail.com | 1 | 2025-12-15 11:23:00 | APPROVED |
| 2 | 2 | user@gmail.com | 3 | 2025-12-15 11:23:17 | BOOKED |
| 3 | 3 | user@gmail.com | 2 | 2025-12-15 11:23:46 | APPROVED |
| 4 | 4 | user@gmail.com | 1 | 2025-12-15 11:56:17 | BOOKED |
| 5 | 5 | user@gmail.com | 1 | 2025-12-15 11:58:52 | BOOKED |

bookings1 x

Apply Revert Context Help Snippets

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|-------------------------------------|--------------------|-----------------------|
| 1 | 18:19:12 | USE travel_db | 0 row(s) affected | 0.000 sec |
| 2 | 18:19:24 | SELECT * FROM bookings LIMIT 0, 300 | 29 row(s) returned | 0.000 sec / 0.000 sec |

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

ENG IN 18:19 17-12-2025

Bookings Table

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variable
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports

Administration Schemas

Information

No object selected

Object Info Session

25* java using sql* OPERATIONS* newproject4(comp.prod)* crud database test* SQL File 11* test* travel_db*

Limit to 300 rows

```

85
86 DESCRIBE bookings;
87
88
89 SHOW TABLES;
90
91 ALTER TABLE bookings

```

Result Grid

Tables_in_travel_db

| | bookings | hotels | travel_package | users |
|----------------|----------|--------|----------------|-------|
| bookings | | | | |
| hotels | | | | |
| travel_package | | | | |
| users | | | | |

Result 3 x

Read Only Context Help Snippets

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|-------------------------------------|--------------------|-----------------------|
| 1 | 18:19:12 | USE travel_db | 0 row(s) affected | 0.000 sec |
| 2 | 18:19:24 | SELECT * FROM bookings LIMIT 0, 300 | 29 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 18:19:40 | DESCRIBE bookings | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 4 | 18:19:50 | SHOW TABLES | 4 row(s) returned | 0.016 sec / 0.000 sec |

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

ENG IN 18:19 17-12-2025

Tables in travel_db

HTTP Request Methods

Definition:

HTTP defines a set of request methods to indicate the action to be performed on a given resource.

| HTTP Method | URI | Purpose |
|-------------|---|-------------------------------|
| GET | http://localhost:8088/view-packages | Fetch all travel packages |
| GET | http://localhost:8088/bookPackage?packageId=3 | Book a selected package |
| GET | http://localhost:8088/my-bookings | Retrieve user booking history |
| POST | http://localhost:8088/doLogin | Login authentication |
| POST | http://localhost:8088/register | Register new user |
| POST | http://localhost:8088/addPackage | Create new travel package |
| POST | http://localhost:8088/addHotel | Create new hotel |
| DELETE | http://localhost:8088/deletePackage?id=7 | Delete travel package |

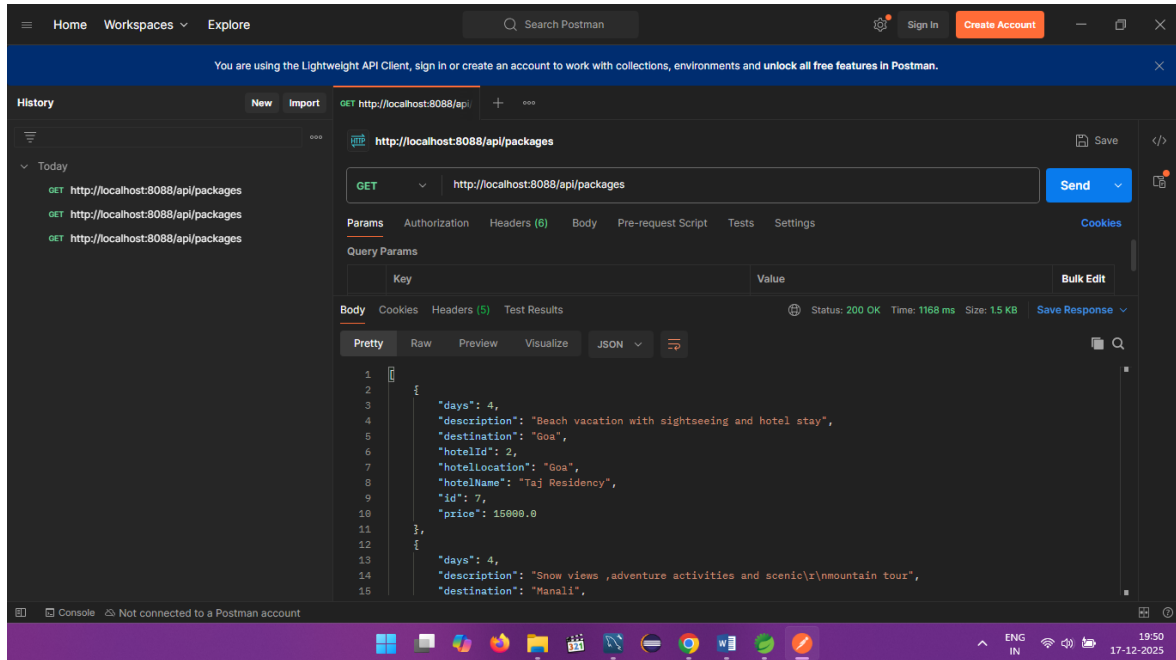
URI (Uniform Resource Identifier)

Definition

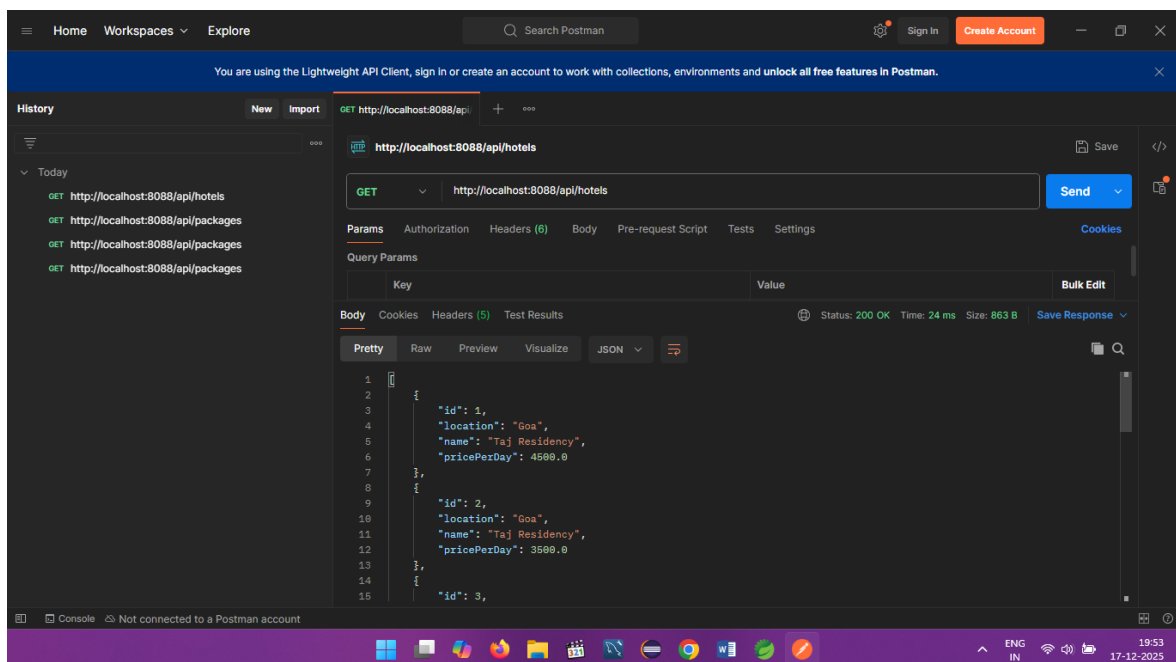
URI is used to uniquely identify resources in a web application. In this project, URI is used to identify users, travel packages, hotels, and bookings.

| HTTP Method | URI | Description |
|-------------|---|--------------------------------------|
| GET | http://localhost:8088/login | Display login page |
| GET | http://localhost:8088/register | Display registration page |
| GET | http://localhost:8088/logout | Logout from system |
| GET | http://localhost:8088/admin-dashboard | Display admin dashboard |
| GET | http://localhost:8088/user-dashboard | Display user dashboard |
| GET | http://localhost:8088/view-packages | View all travel packages |
| GET | http://localhost:8088/my-bookings | View logged-in user bookings |
| GET | http://localhost:8088/add-package | Open add travel package page (Admin) |
| POST | http://localhost:8088/addPackage | Add new travel package |
| GET | http://localhost:8088/add-hotel | Open add hotel page (Admin) |
| POST | http://localhost:8088/addHotel | Add new hotel |
| GET | http://localhost:8088/bookPackage?packageId=5 | Book a travel package |
| POST | http://localhost:8088/admin/update-booking | Approve/Reject booking |

POSTMAN DEMONSTRATION

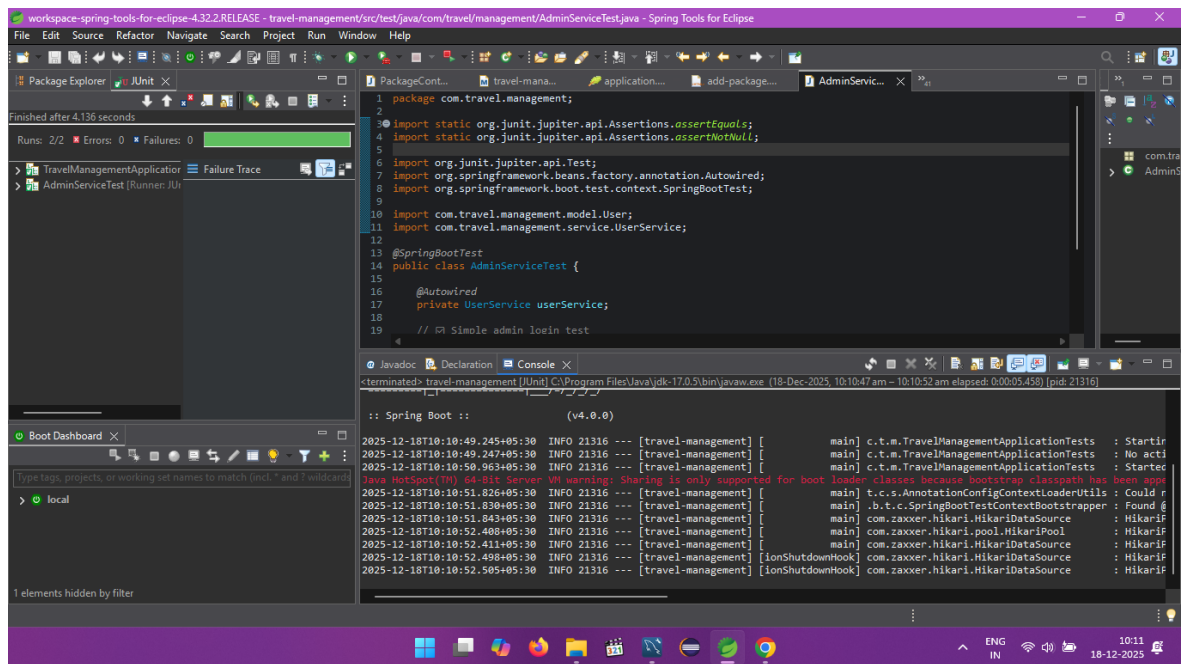


Get Method for packages

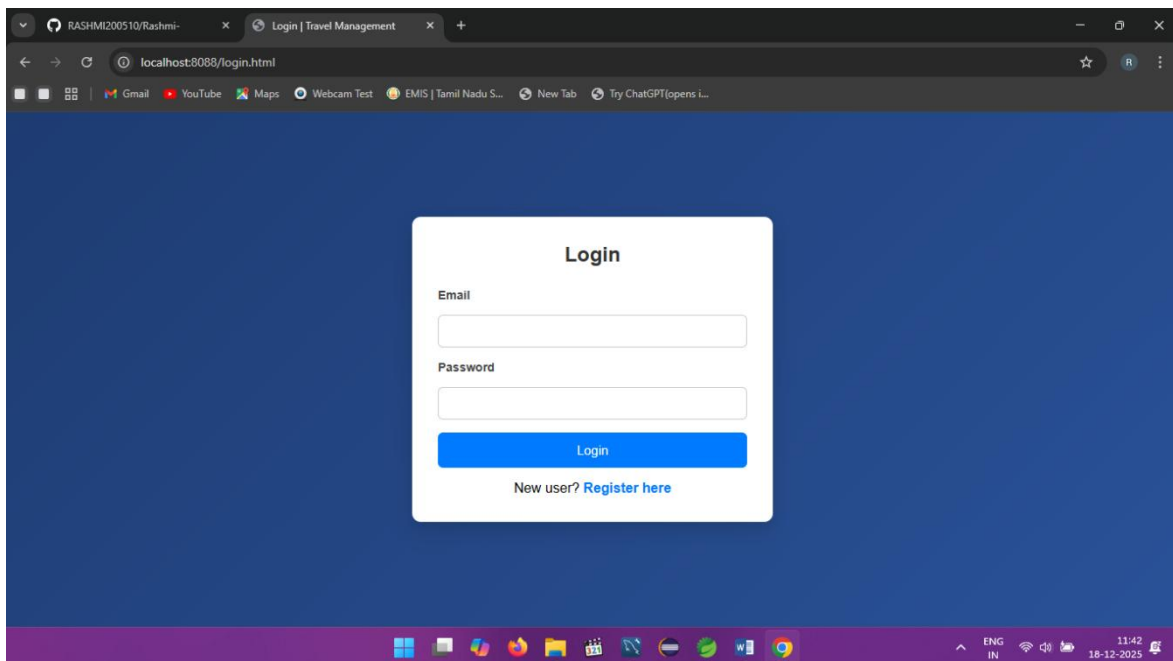


Get method for hotels

JUNIT TEST FOR ADMIN MODULE



PROJECT OUTPUTS



Login Page

Registration Page

A screenshot of a web browser window displaying a registration page. The browser's address bar shows the URL `localhost:8088/register.html`. The page has a dark blue background. In the center, there is a white rectangular form titled "Register". The form contains four input fields: "Name", "Email", "Password", and "Confirm Password". Below these fields is a blue "Register" button. At the bottom of the form, there is a link that says "Already have an account? [Login](#)". The browser's taskbar at the bottom shows various application icons and the system clock indicating 11:45 on 18-12-2025.

A screenshot of the same web browser window, but now the registration form is filled with sample data. The "Name" field contains "Reena", the "Email" field contains "reena@gmail.com", the "Password" field contains "*****", and the "Confirm Password" field also contains "*****". The blue "Register" button remains visible below the fields. The "Login" link is still present at the bottom of the form. The browser's taskbar and system clock are consistent with the previous screenshot, showing 11:46 on 18-12-2025.

User Login

Login

Email
reena@gmail.com

Password
.....

Login

New user? [Register here](#)

User Dashboard

Travel Management Logged in as: User [Logout](#)

Dashboard

Welcome User
Your travel summary is shown below.

0
Total Bookings

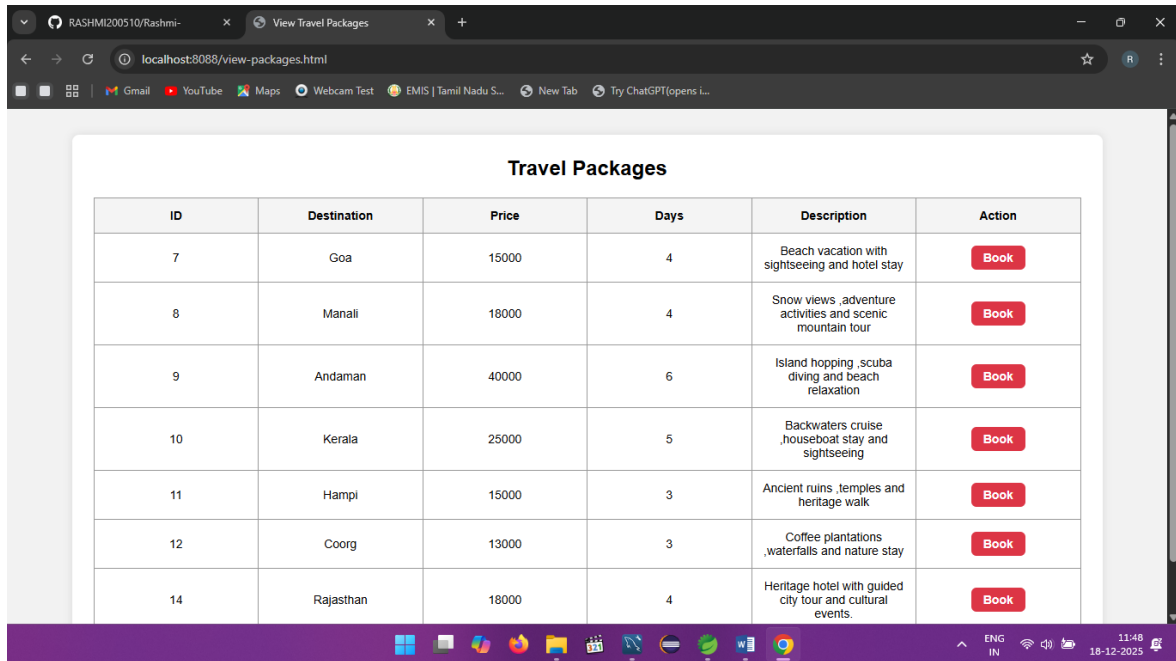
0
Approved Trips

0
Pending Approval

Approved Travel Receipts

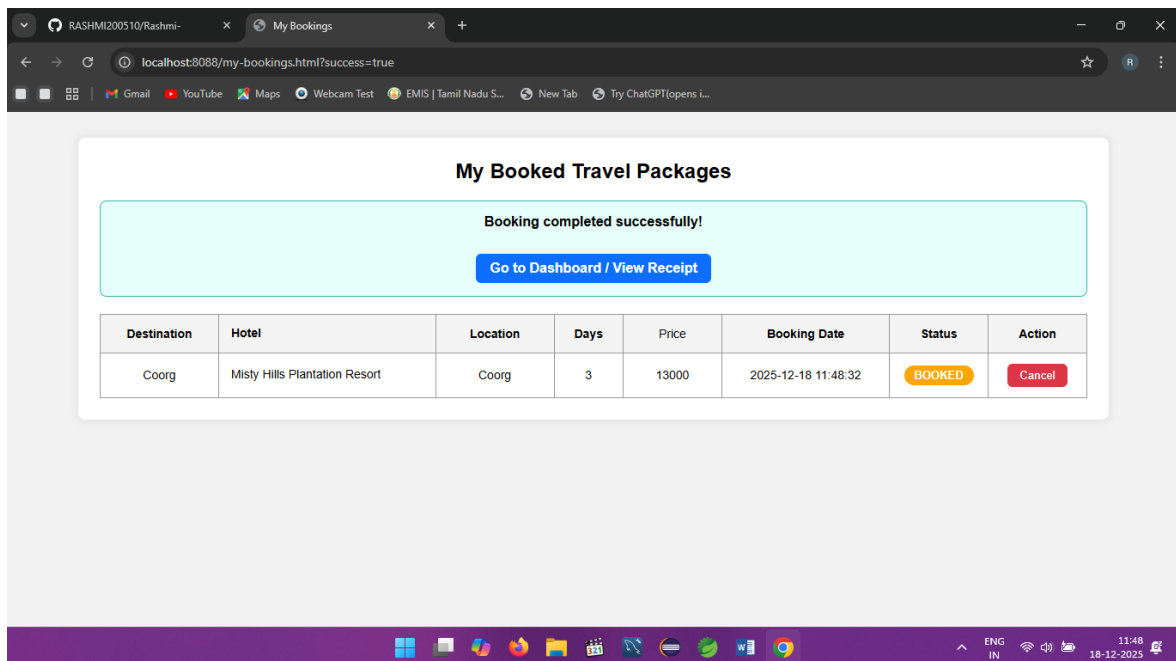
| Destination | Days | Price | Booking Date | Status |
|-----------------------|------|-------|--------------|--------|
| No approved trips yet | | | | |

Travel Packages



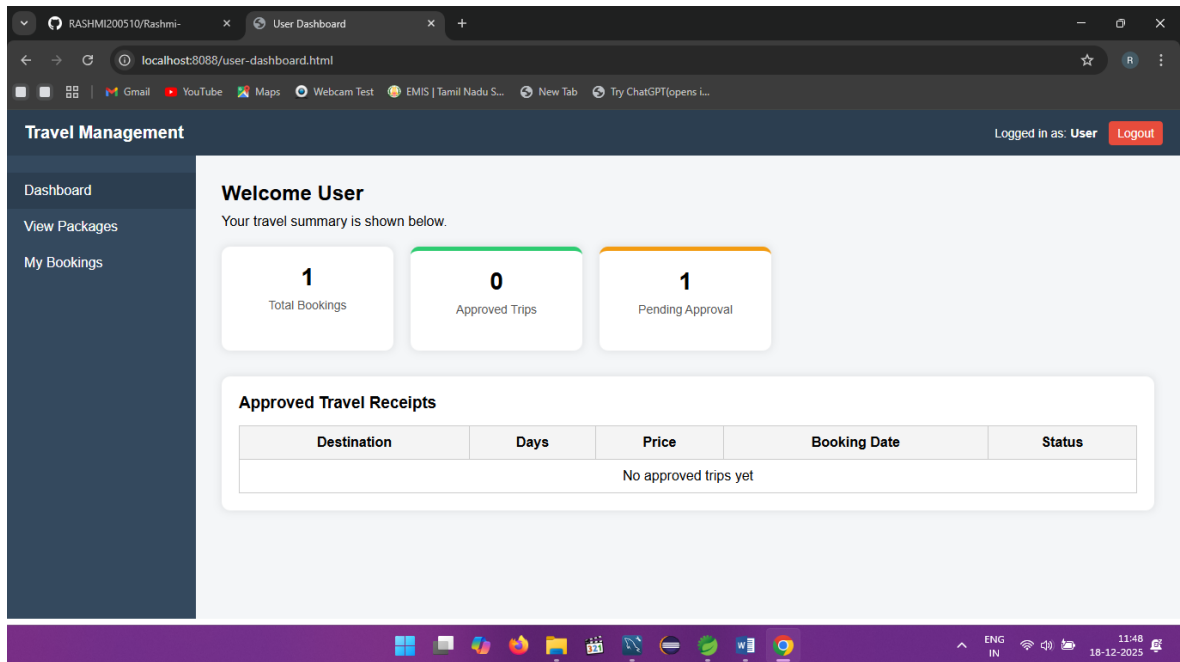
| ID | Destination | Price | Days | Description | Action |
|----|-------------|-------|------|---|----------------------|
| 7 | Goa | 15000 | 4 | Beach vacation with sightseeing and hotel stay | Book |
| 8 | Manali | 18000 | 4 | Snow views, adventure activities and scenic mountain tour | Book |
| 9 | Andaman | 40000 | 6 | Island hopping, scuba diving and beach relaxation | Book |
| 10 | Kerala | 25000 | 5 | Backwaters cruise, houseboat stay and sightseeing | Book |
| 11 | Hampi | 15000 | 3 | Ancient ruins, temples and heritage walk | Book |
| 12 | Coorg | 13000 | 3 | Coffee plantations, waterfalls and nature stay | Book |
| 14 | Rajasthan | 18000 | 4 | Heritage hotel with guided city tour and cultural events | Book |

My Booked Travel Packages



| Booking completed successfully! | | | | | | | |
|--|-------------------------------|----------|------|-------|---------------------|--------|------------------------|
| Go to Dashboard / View Receipt | | | | | | | |
| Destination | Hotel | Location | Days | Price | Booking Date | Status | Action |
| Coorg | Misty Hills Plantation Resort | Coorg | 3 | 13000 | 2025-12-18 11:48:32 | BOOKED | Cancel |

User Dashboard After Booking



The screenshot shows a web browser window with the URL `localhost:8088/user-dashboard.html`. The page is titled "Travel Management" and shows the user is logged in as "User" with a "Logout" button. The dashboard includes a sidebar with "Dashboard", "View Packages", and "My Bookings". The main content area displays a "Welcome User" message and a travel summary with three cards: "1 Total Bookings", "0 Approved Trips", and "1 Pending Approval". Below this is a section for "Approved Travel Receipts" with a table that currently shows "No approved trips yet".

Travel Management Logged in as: User [Logout](#)

Dashboard
View Packages
My Bookings

Welcome User
Your travel summary is shown below.

1
Total Bookings

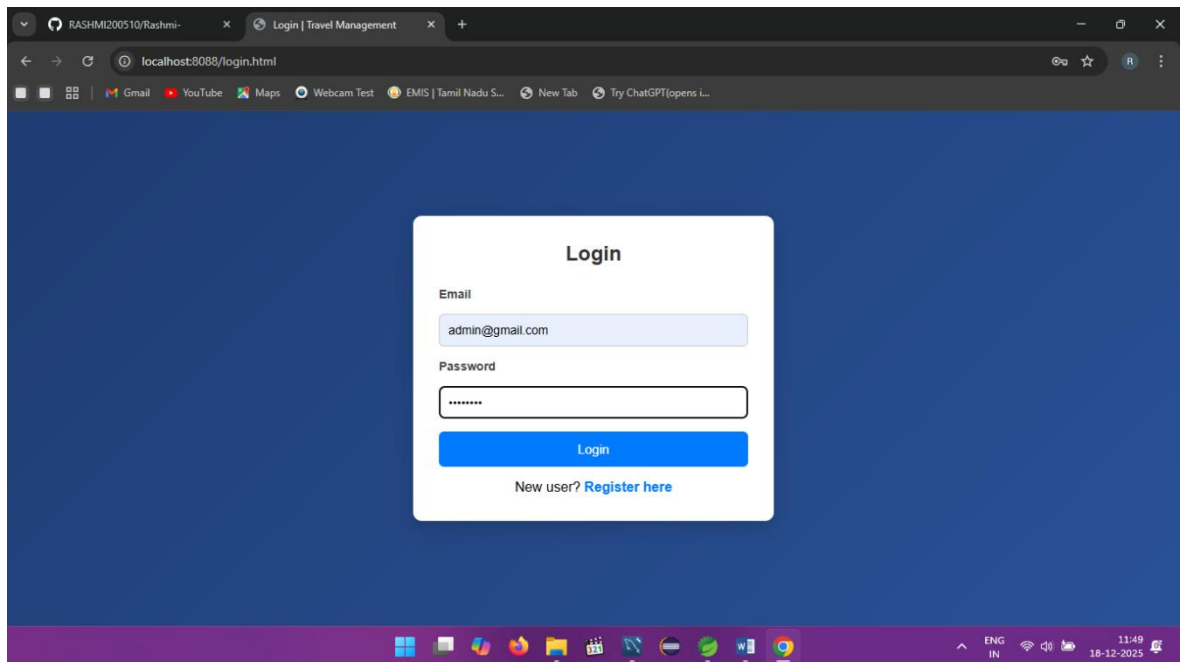
0
Approved Trips

1
Pending Approval

Approved Travel Receipts

| Destination | Days | Price | Booking Date | Status |
|-----------------------|------|-------|--------------|--------|
| No approved trips yet | | | | |

Admin Login



The screenshot shows a web browser window with the URL `localhost:8088/login.html`. The page has a blue background and a white login form. The form includes fields for "Email" (with the value `admin@gmail.com`) and "Password" (masked with asterisks). There is a blue "Login" button and a link for "New user? Register here".

Login

Email
`admin@gmail.com`

Password

[Login](#)

New user? [Register here](#)

Admin Dashboard

Travel Management Logged in as: Admin [Logout](#)

Dashboard Overview

- Total Bookings: **33**
- Total Packages: **7**
- Total Hotels: **9**
- Total Users: **6**

Pending Bookings

| User | Destination | Hotel | Days | Price | Status | Action |
|-----------------|-------------|-------------------------------|------|-------|--------|--|
| reena@gmail.com | Coorg | Misty Hills Plantation Resort | 3 | 13000 | BOOKED | Approve Reject |

Bookings Page of Admin

| User | Destination | Days | Price | Date | Status | Action |
|-------------------|-------------|------|-------|---------------------|-----------|--------|
| user@gmail.com | Coorg | 3 | 13000 | 2025-12-16 12:03:33 | APPROVED | - |
| user@gmail.com | Manali | 4 | 18000 | 2025-12-16 12:24:23 | APPROVED | - |
| user@gmail.com | Hampi | 3 | 15000 | 2025-12-16 14:33:38 | CANCELLED | - |
| user@gmail.com | Andaman | 6 | 40000 | 2025-12-16 18:28:56 | APPROVED | - |
| user@gmail.com | Goa | 4 | 15000 | 2025-12-16 19:42:43 | APPROVED | - |
| user@gmail.com | Manali | 4 | 18000 | 2025-12-16 22:20:50 | APPROVED | - |
| user@gmail.com | Kerala | 5 | 25000 | 2025-12-16 22:22:56 | APPROVED | - |
| kashika@gmail.com | Goa | 4 | 15000 | 2025-12-16 22:27:02 | APPROVED | - |
| user@gmail.com | Kerala | 5 | 25000 | 2025-12-17 19:58:24 | APPROVED | - |
| user@gmail.com | Goa | 4 | 15000 | 2025-12-17 20:41:49 | APPROVED | - |
| harshi@gmail.com | Kerala | 5 | 25000 | 2025-12-18 10:17:02 | APPROVED | - |
| reena@gmail.com | Coorg | 3 | 13000 | 2025-12-18 11:48:32 | APPROVED | - |

Updated Admin Dashboard

The screenshot shows a web browser window with the URL `localhost:8088/admin-dashboard.html`. The page is titled "Travel Management" and shows the user is logged in as "Admin" with a "Logout" button. The sidebar on the left contains the following links: Dashboard, Travel Packages, Add Package, and Add Hotel. The main content area is titled "Dashboard Overview" and displays four summary cards: Total Bookings (33), Total Packages (7), Total Hotels (9), and Total Users (6). Below these cards is a section titled "Pending Bookings" which contains a table with columns: User, Destination, Hotel, Days, Price, Status, and Action. The table currently shows "No pending bookings".

Travel Management Logged in as: Admin [Logout](#)

Dashboard Overview

- Total Bookings: 33
- Total Packages: 7
- Total Hotels: 9
- Total Users: 6

Pending Bookings

| User | Destination | Hotel | Days | Price | Status | Action |
|---------------------|-------------|-------|------|-------|--------|--------|
| No pending bookings | | | | | | |

Add Travel Packages Page

The screenshot shows a web browser window with the URL `localhost:8088/add-package.html`. The page is titled "Add Travel Package" and contains a form with the following fields: Destination (Goa), Price (15000), Days (4), Description (Beach vacation with sightseeing and hotel stay), and Hotel (Taj Residency (Goa)). A blue "Add Package" button is at the bottom of the form.

Add Travel Package

Destination
Goa

Price
15000

Days
4

Description
Beach vacation with sightseeing and hotel stay

Hotel
Taj Residency (Goa)

[Add Package](#)

Add Hotel Page

Add Hotel

Hotel Name
Taj Residency

Location
Goa

Price Per Day
4500

Add Hotel

User Dashboard After Approval

Travel Management Logged in as: User [Logout](#)

Dashboard

Welcome User
Your travel summary is shown below.

1
Total Bookings

1
Approved Trips

0
Pending Approval

Approved Travel Receipts

| Destination | Days | Price | Booking Date | Status |
|-------------|------|-------|---------------------|-----------------|
| Coorg | 3 | 13000 | 2025-12-18 11:48:32 | APPROVED |