**A**

**Project Report**

**On**

Tourism Management System

**By :**

Meet Vekariya

Semester – 5

**Project Guide :**

PROF. JAYTIKA JOGIYA

**Submitted To :**

Geetanjali College Of Computer

Science And Commerce ( B.B.A ). Rajkot.

**Academic Year :** 2024-2025

**ACKNOWLEDGEMENT**

I Am Happy To Submit My Idea Of " Tourism Management System" Application

In Saurashtra University, Rajkot For BCA Degree In Computer Branch.

We take this occasion to thank God, almighty for blessing us with his grace and

taking our Endeavour to a successful culmination. We extend our sincere and

heartfelt thanks to our esteemed guide, PROF. JAYTIKA JOGIYA for providing us

with the right guidance and advice at the crucial junctures and for showing us

the right way. We would like to thank the other faculty members also, at this

occasion. Last but not the least, we would like to thank friends for the support

and encouragement they have given us during the course of our work.

INTRODUCTION

System Development:

The process of building systems has always been complex with system becoming larger, the costs and complexities get multiplied. So the need for better methods for developing systems is widely recognized to be effective and the applied model should meet a few basic requirements.

The model should be structured and cover the entire system development process from feasibility study to programming, testing and implementation.

The model should utilize established methods and techniques like database designs, normalizations and structured programming techniques.

The model should consist of building blocks, which define tasks, results and interfaces.

The model should separate the logical system from the physical system.

Documentation should be a direct result of the development work and should be concise, precise and as non-redundant as possible.

Based on the above requirements of the system model, system study has been made. Various methodologies have been applied for system study, evolving design documents, data modeling, input screen design and report design.

**Project:**

The **Tourism Management System** is designed to streamline and enhance the experience of tourists and travellers by providing an integrated platform for managing all aspects of travel planning. As tourism continues to grow into one of the most dynamic sectors globally, there is an increasing demand for efficient tools that facilitate smooth travel experiences—from finding destinations and booking hotels to connecting with local guides and arranging travel packages.

This system serves as a comprehensive solution for tourists, offering them the ability to search for destinations, hotels, and tour packages, and make bookings all in one place. In addition, it empowers travel administrators to easily manage destinations, accommodations, and guides, ensuring that up-to-date information is always available to the user.

Developed using **PHP** and **MySQL**, the Tourism Management System leverages modern web technologies to offer a seamless user experience. It incorporates two primary modules: the **Admin Module**, which allows for the management of travel resources, and the **Tourist Module**, which allows users to explore and book their ideal travel experiences.

The key motivation behind this project is to simplify the complexity often associated with travel planning, making it more accessible and enjoyable for tourists while providing travel managers and service providers with a robust, easy-to-use platform to manage their offerings.

We hope this system will serve as a valuable tool for both tourists seeking to discover new places and travel administrators working to create memorable experiences for travellers.

index

**Functional Requirements:**

**Admin Module:**

**Dashboard :** In this section, admin can view the details in brief like total tour

listed, tour book requests and message from contact us.

**Tour Manage:** In this section, admin can manage tour packages

(Add/Update/Delete).

**Tour Book :** In this section, admin can confirm or delete request from user

Booking tour package.

**Contact Manage :** In this section, admin can view a message from user.

Admin can also update his profile, change the password and recover the password.

**User Module :**

**Home:** Its is welcome page for users and tourist. If any users want to travel the

India they must register with us.

**Tour Packages:** User can book a tour from this page.

**About Us:** Users can view the about us page.

**Contact Us:** Users can contact with admin the through contact us page.

**Registered Users:**

**Home:** Its is welcome page for users and tourist. If any users want to travel the

India they must register with us.

**Tour Packages:** User can book a tour from this page.

**About Us:** Users can view the about us page.

**Contact Us:** Users can contact with admin the through contact us page.

**My Account:**

* Profile
* Logout

User can also update his profile, change the password and recover the password.

**Non-Functional Requirements:**

**Performance:**

* The system must handle multiple concurrent users (tourists and admins) without performance degradation.

**Scalability:**

* The system should support a growing number of destinations and packages.
* The architecture should allow easy scaling by adding more servers or database instances.

**Security:**

* Secure login and authentication for both admin and tourists.
* Data encryption for sensitive information (user passwords).
* Role-based access control for admin and tourist modules.
* Protection against common web vulnerabilities like SQL injection, XSS, and CSRF.

**Availability:**

* The system should be available 24/7 with minimal downtime for updates or maintenance.

**Usability:**

* The user interface should be intuitive and easy to navigate for both tourists and admins.
* Consistent design and layout for all pages and functionalities.

**Compatibility:**

* The system should be responsive and accessible from various devices, including desktops, laptops, tablets, and smartphones.
* Cross-browser compatibility (Chrome, Firefox, Safari, Edge, etc.).

**Maintainability:**

* Well-structured and documented code to enable future maintenance and feature extensions.
* Modular design for easy updates and bug fixes.

**Technical Requirements:**

**Front-End Technology:**

* **HTML5, CSS3, JavaScript**: For creating the user interface (UI).
* **Bootstrap**: For responsive design and consistent UI across devices.
* **AJAX**: For dynamic data loading without refreshing pages.

**Back-End Technology:**

* **PHP**: For server-side programming and handling requests.
* **MySQL**: For managing the database, including storage of destinations, packages, bookings, users.
* **Apache**: For web server management.

**Database Design:**

* **Tables**:
  + Users
  + Admin
  + Bookedpkg
  + Contactus
  + Packages
  + Bookings

**Data Dictionary**

**Admin Table :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Collation** | **Null** | **Default** | **Extra** |
| 1 | **Id** | Int(11) |  | No | None | AUTO\_INCREMENT |
| 2 | **Username** | Varchar(50) | utf8mb4\_general\_ci | No | None |  |
| 3 | **MobileNumber** | Varchar(50) | utf8mb4\_general\_ci | No | None |  |
| 4 | **Email** | Varchar(11) | utf8mb4\_general\_ci | No | None |  |
| 5 | **Password** | Varchar(50) | utf8mb4\_general\_ci | No | None |  |

**Bookedpkg:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Collation** | **Null** | **Default** | **Extra** |
| 1 | **id** | Int |  | No | None | AUTO\_INCREMENT |
| 2 | **package\_id** | int |  | No | None |  |
| 3 | **booking\_id** | int |  | No | None |  |
| 4 | **tour\_name** | varchar(255) | utf8mb4\_0900\_ai\_ci | No | None |  |
| 5 | **tour\_date** | varchar(255) | utf8mb4\_0900\_ai\_ci | No | None |  |
| 6 | **status** | enum('pending', 'confirm') | utf8mb4\_0900\_ai\_ci | No | None |  |

**Booking:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Collation** | **Null** | **Default** | **Extra** |
| 1 | **id** | int |  | No | None | AUTO\_INCREMENT |
| 2 | **package\_id** | int |  | No | None |  |
| 3 | **name** | varchar(255) | utf8mb4\_general\_ci | No | None |  |
| 4 | **email** | varchar(255) | utf8mb4\_general\_ci | No | None |  |
| 5 | **phone** | varchar(50) | utf8mb4\_general\_ci | No | None |  |
| 6 | **status** | enum('pending', 'confirmed') | utf8mb4\_general\_ci | No | None |  |
| 7 | **created\_at** | timestamp |  | No | None |  |

**Contact us:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Collation** | **Null** | **Default** | **Extra** |
| 1 | **id** | int |  | No | None | AUTO\_INCREMENT |
| 2 | **name** | varchar(50) | utf8mb4\_general\_ci | No | None |  |
| 3 | **email** | varchar(50) | utf8mb4\_general\_ci | No | None |  |
| 4 | **subject** | varchar(50) | utf8mb4\_general\_ci | No | None |  |
| 5 | **message** | varchar(200) | utf8mb4\_general\_ci | No | None |  |
| 6 | **datetime** | timestamp |  | No | None |  |

**Package:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Collation** | **Null** | **Default** | **Extra** |
| 1 | | **id** | int |  | No | None | AUTO\_INCREMENT |
| 2 | **location** | varchar(100) | utf8mb4\_general\_ci | No | None |  |
| 3 | **days** | int |  | No | None |  |
| 4 | **persons** | int |  | No | None |  |
| 5 | **rating** | float |  | No | None |  |
| 6 | **price** | decimal(10,2) |  | No | None |  |
| 7 | **image\_url** | varchar(255) | utf8mb4\_general\_ci | No | None |  |

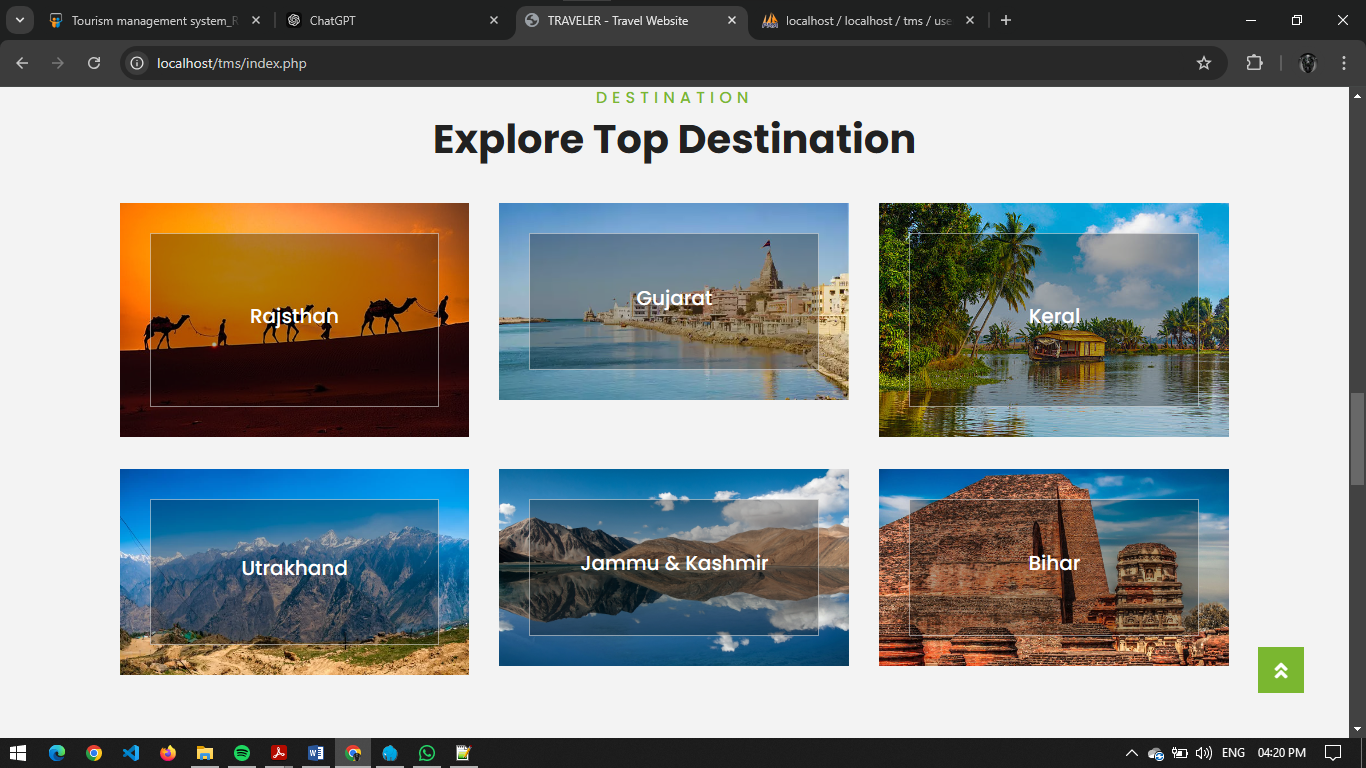
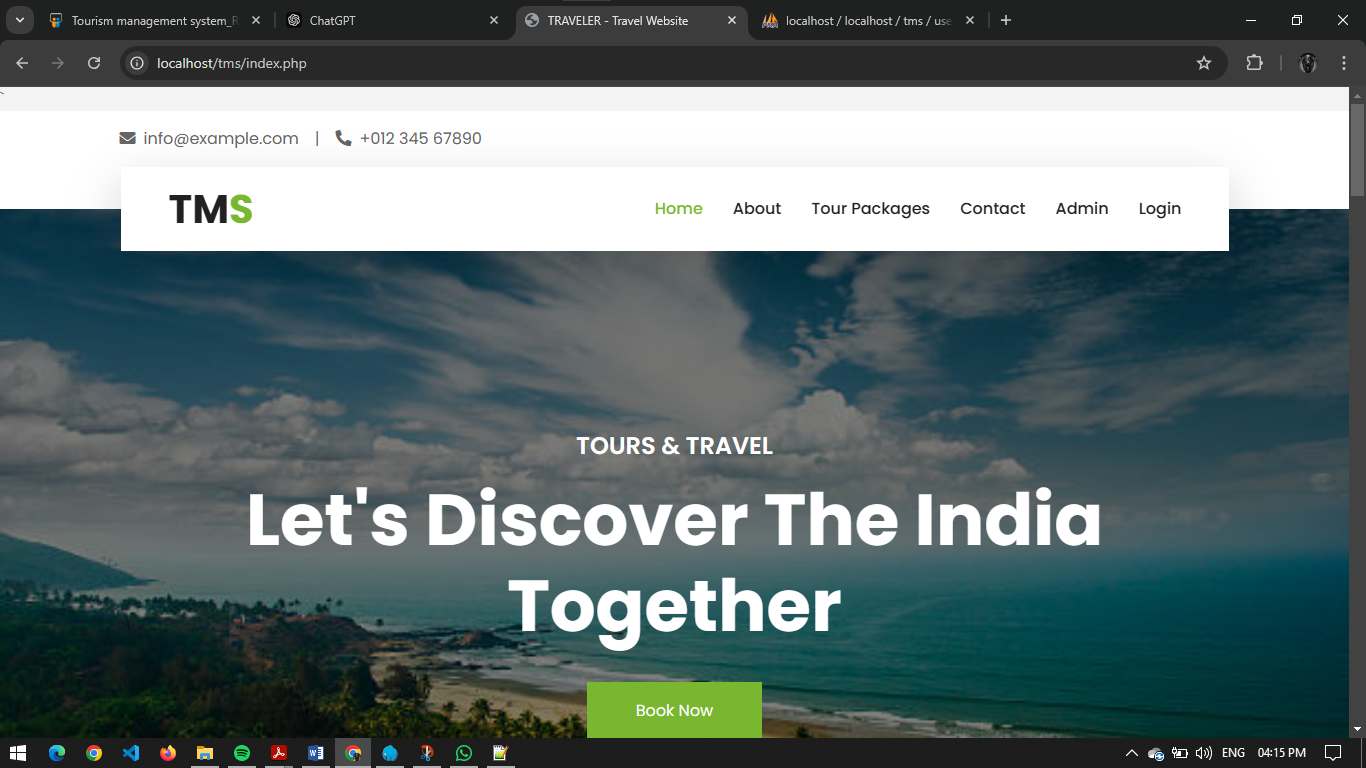
**Users:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Collation** | **Null** | **Default** | **Extra** |
| 1 | **id** | int |  | No | None | AUTO\_INCREMENT |
| 2 | **full\_name** | varchar(100) | utf8mb4\_general\_ci | No | None |  |
| 3 | **email** | varchar(100) | utf8mb4\_general\_ci | No | None |  |
| 4 | **phone** | varchar(15) | utf8mb4\_general\_ci | No | None |  |
| 5 | **password** | varchar(255) | utf8mb4\_general\_ci | No | None |  |

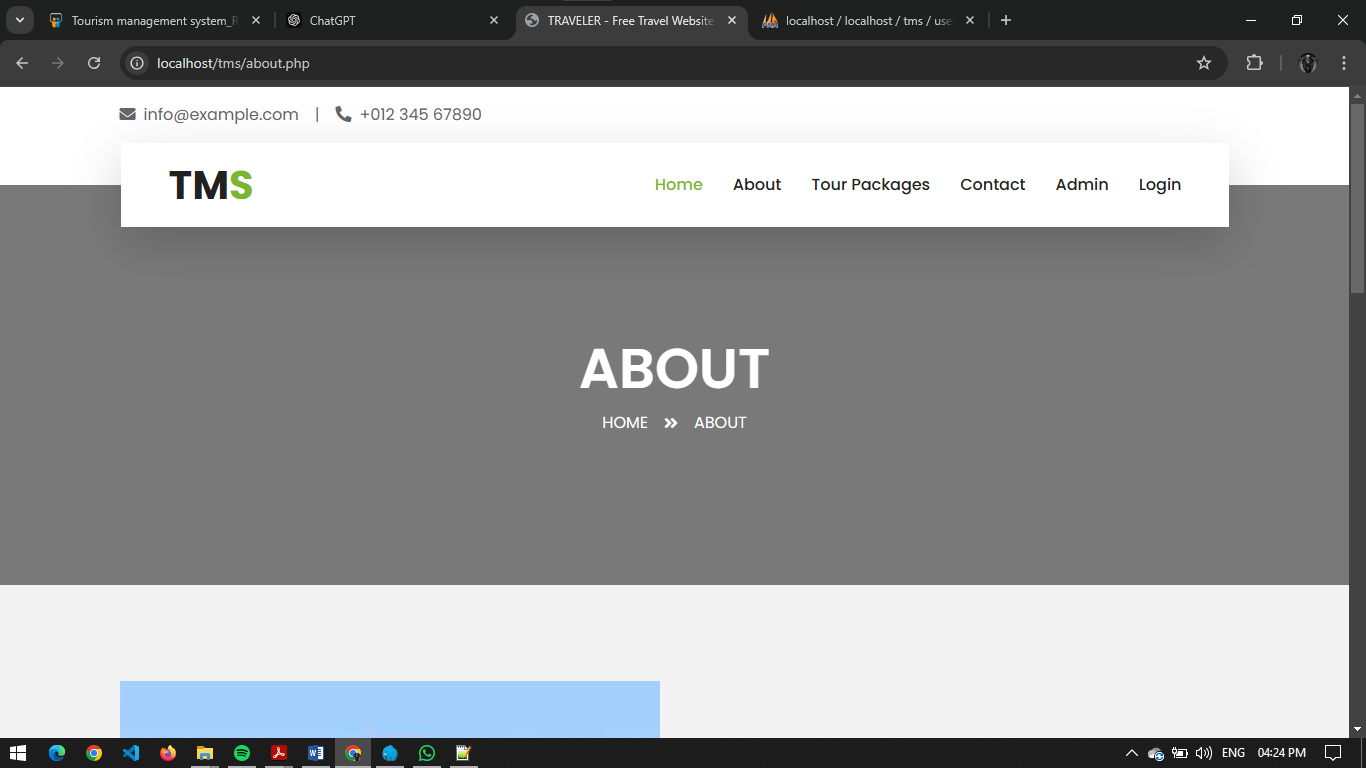
**Project\_Screenshots**

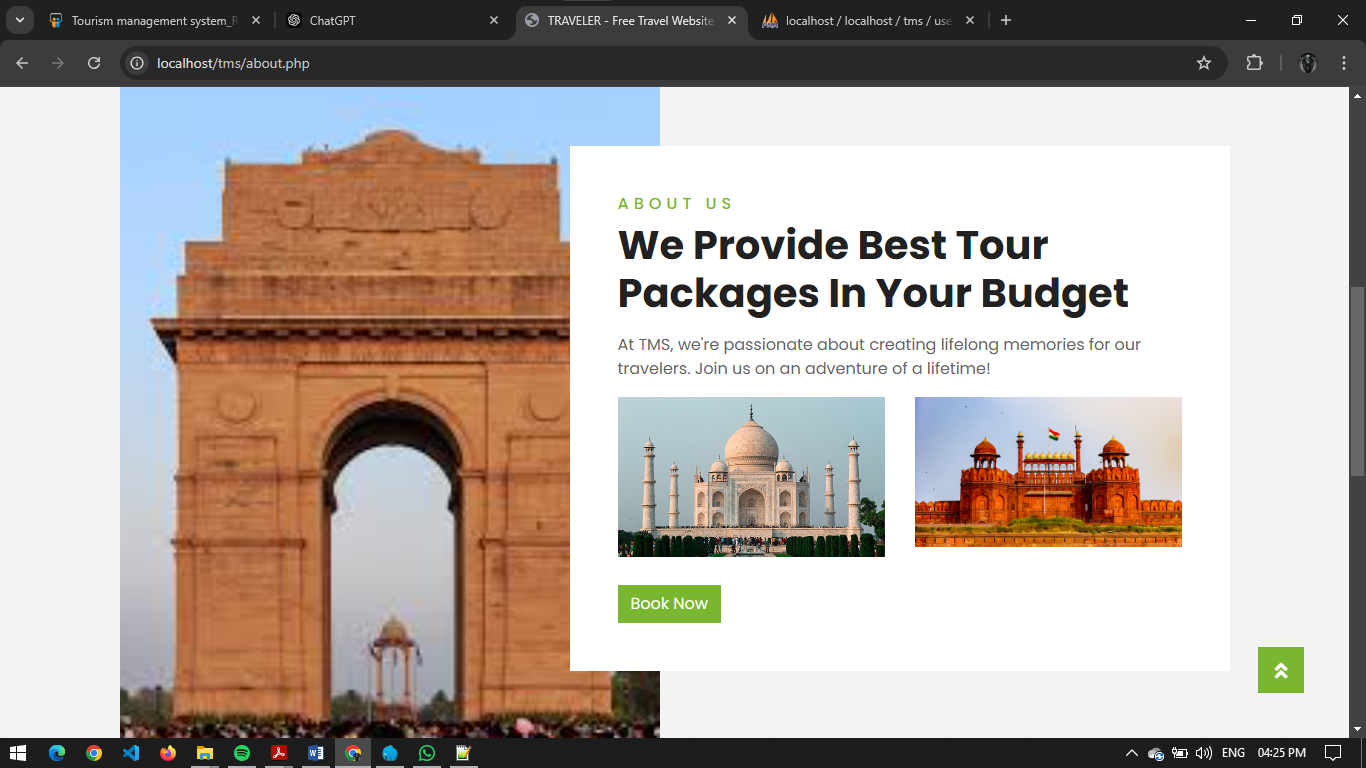
**User Panel:**

**Home page:**

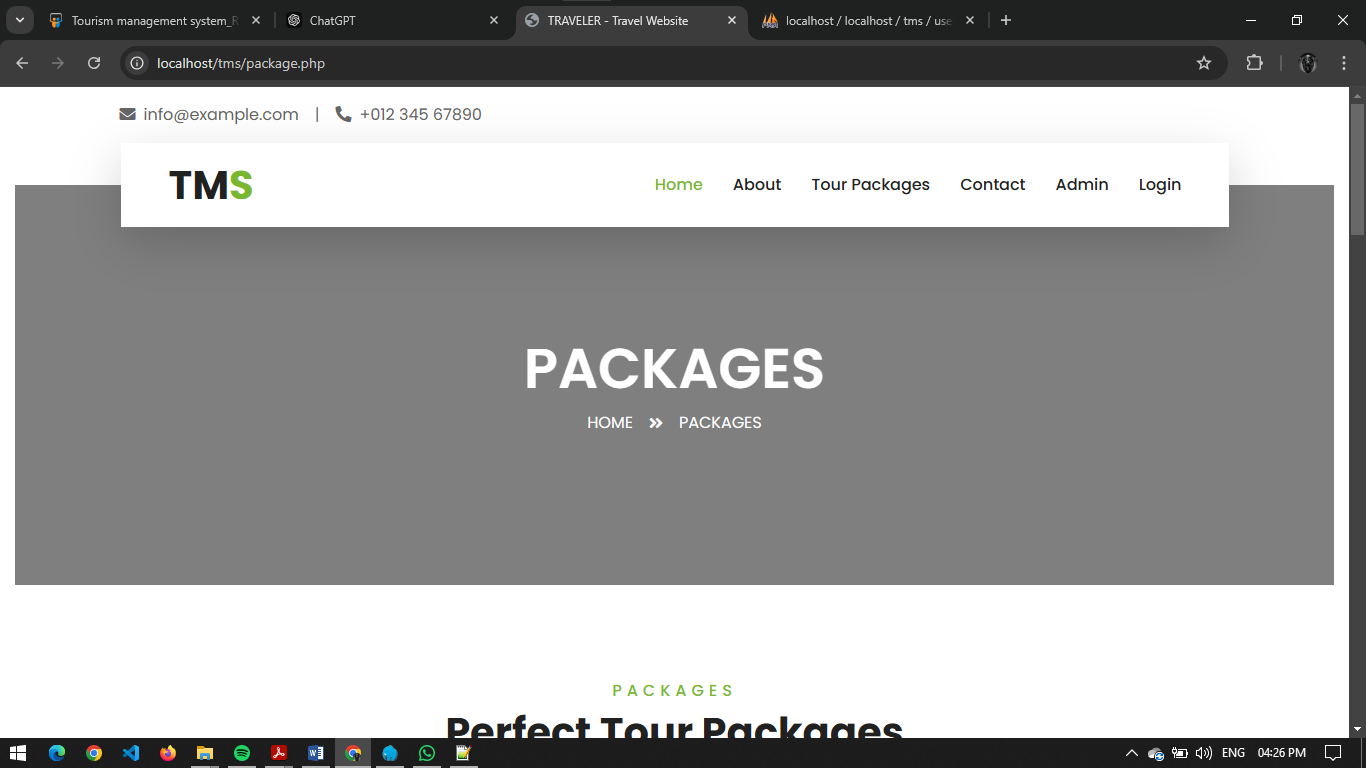
****

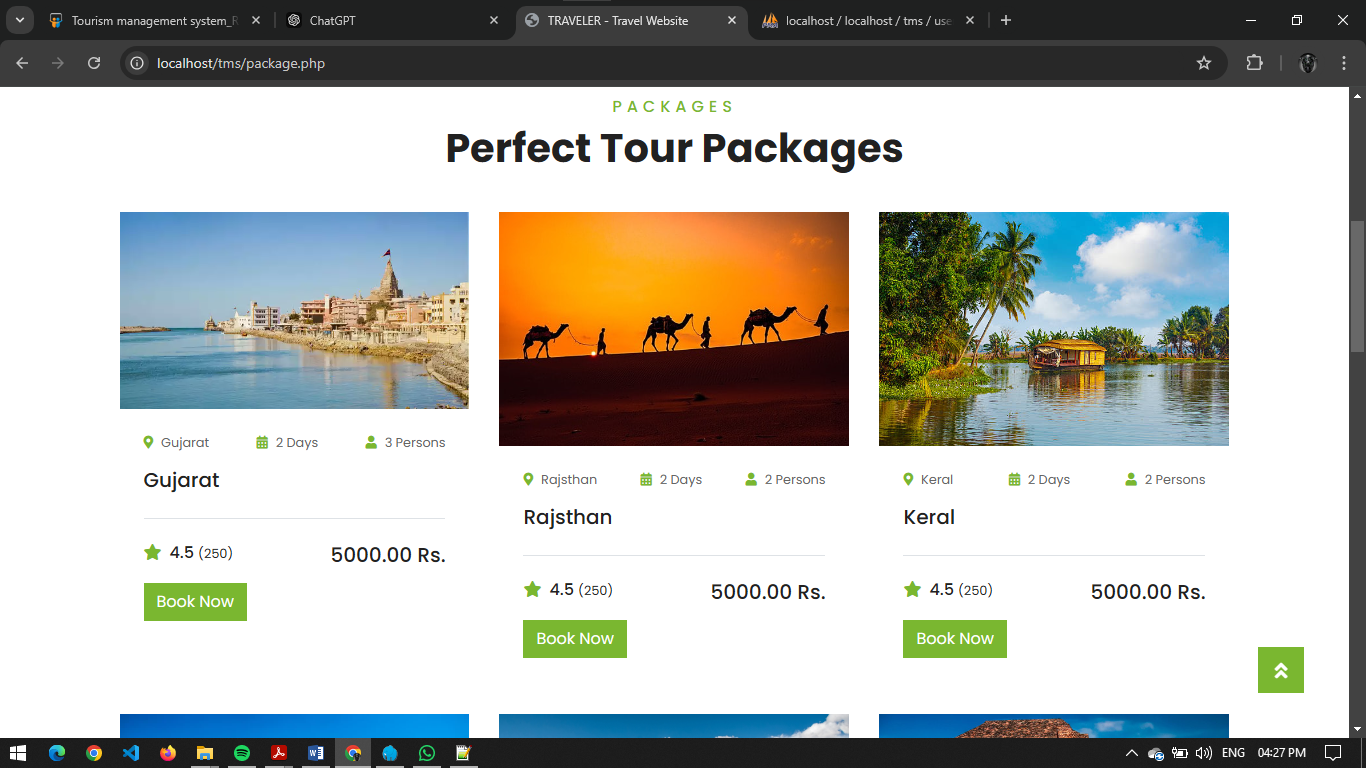
**About Us:**

****

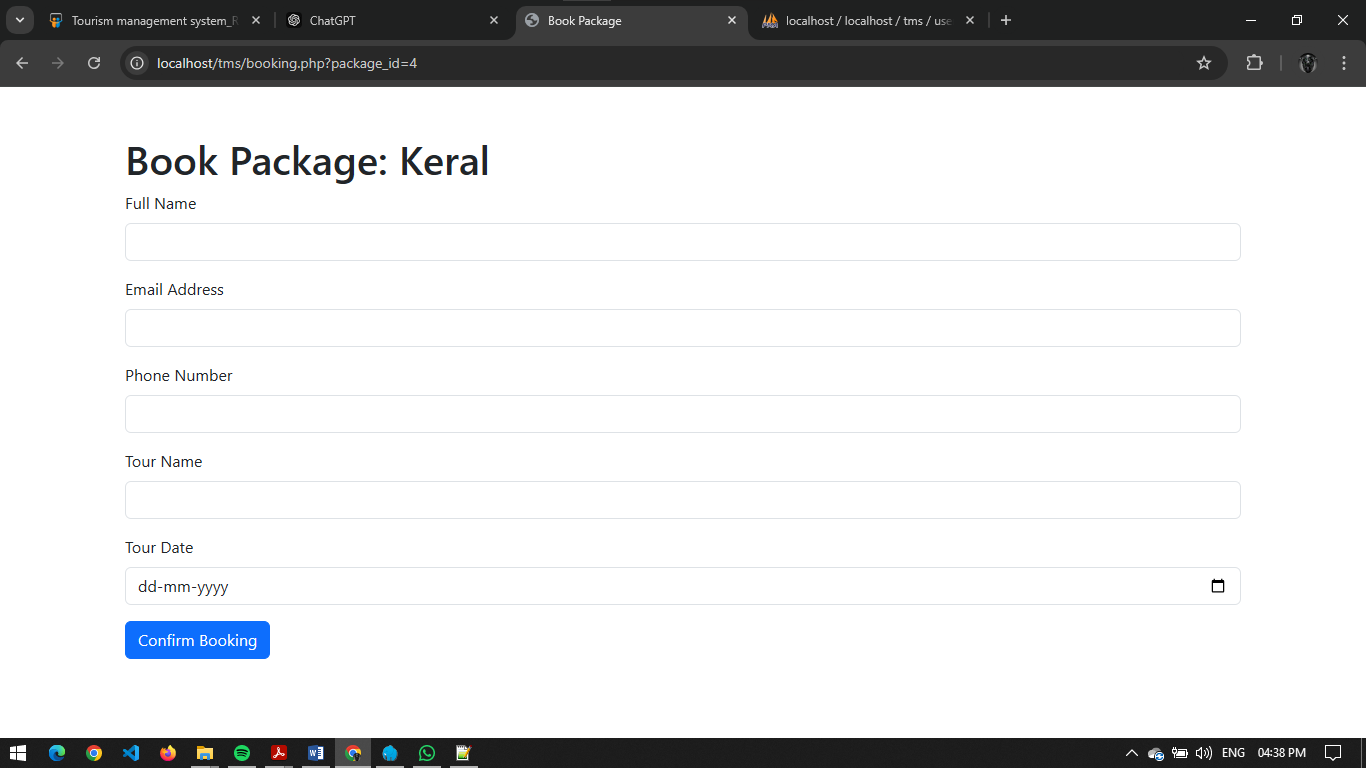


**Tour Package:**

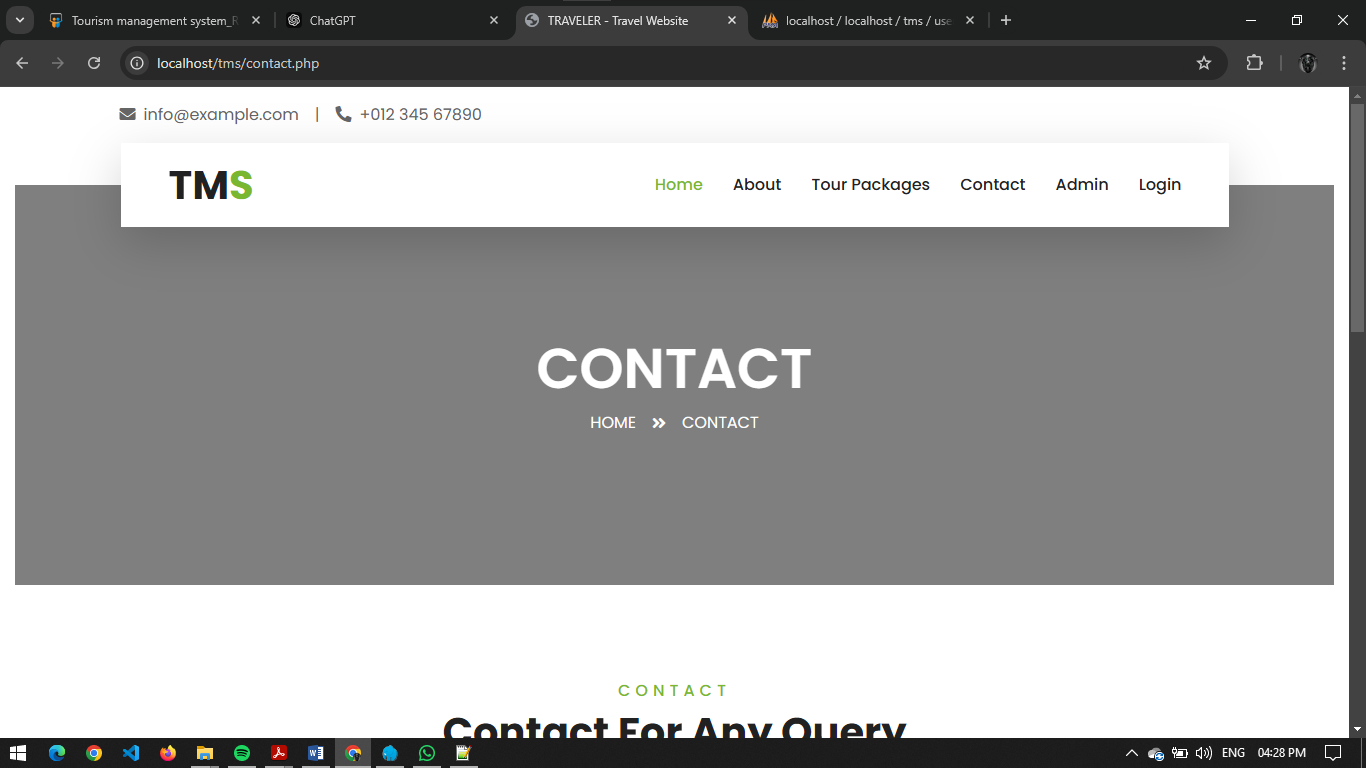
****

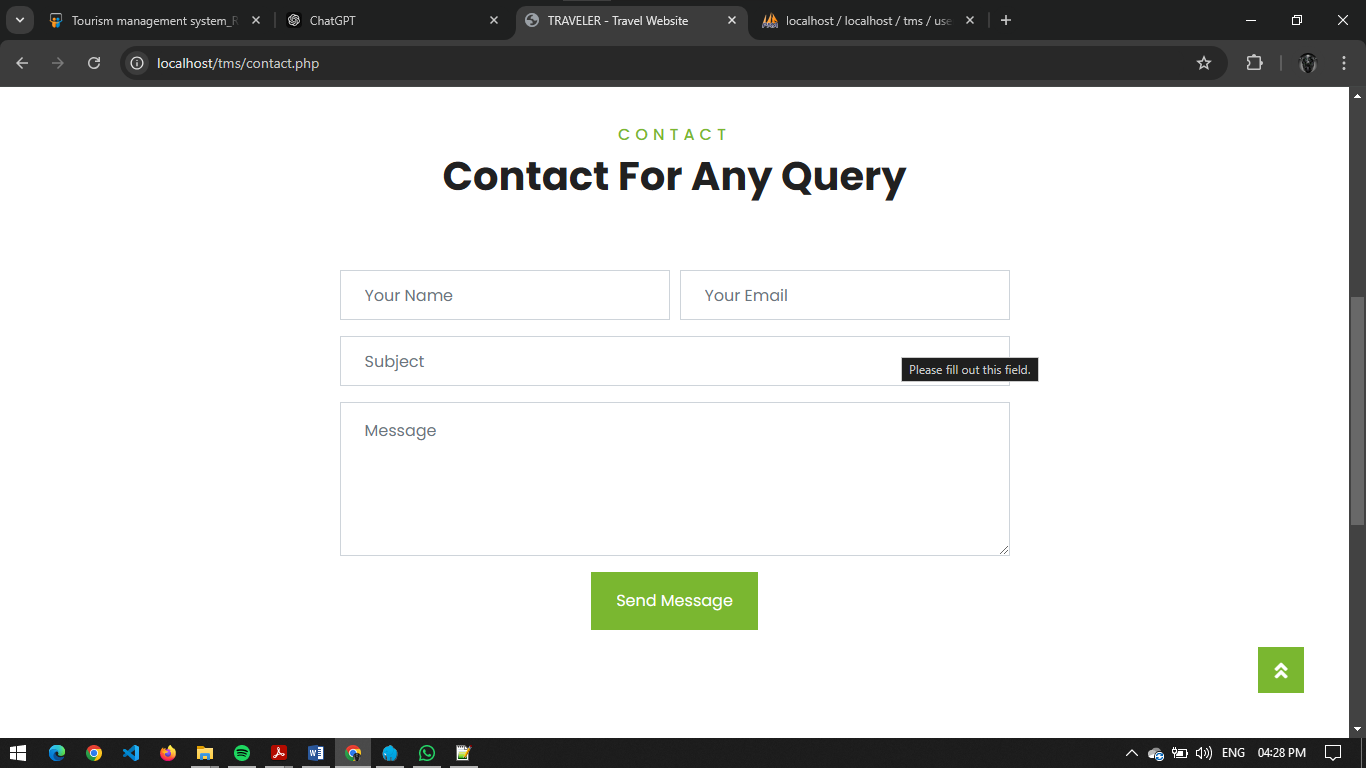


**Package Booking:**

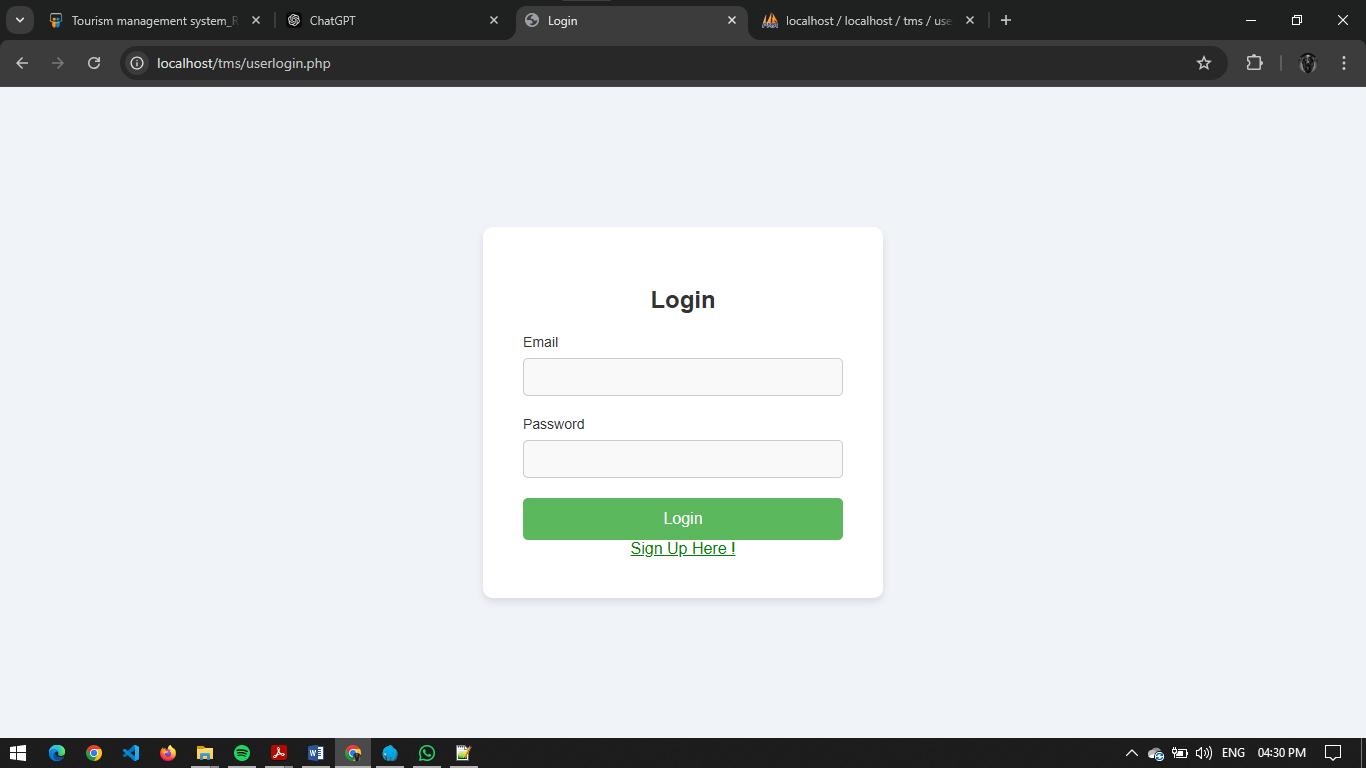
****

**Contact Us:**

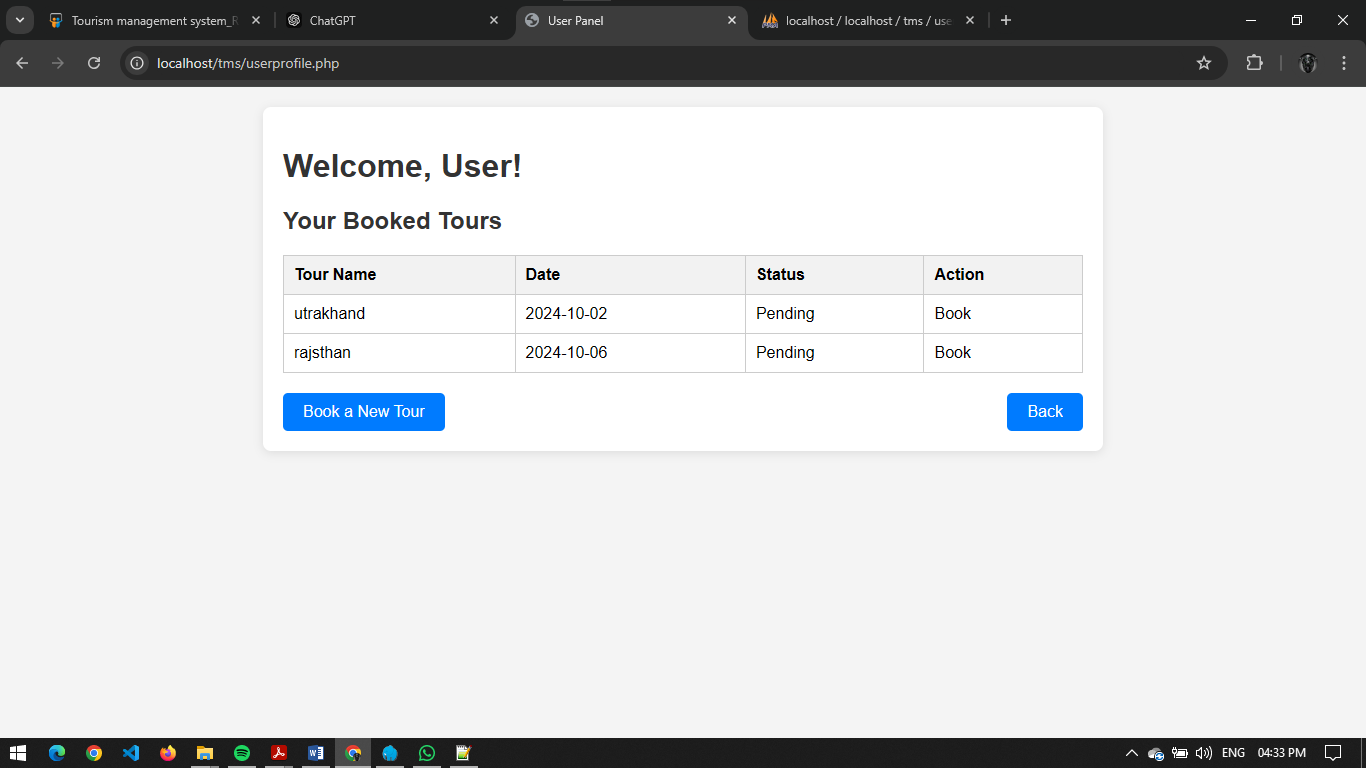
****

****

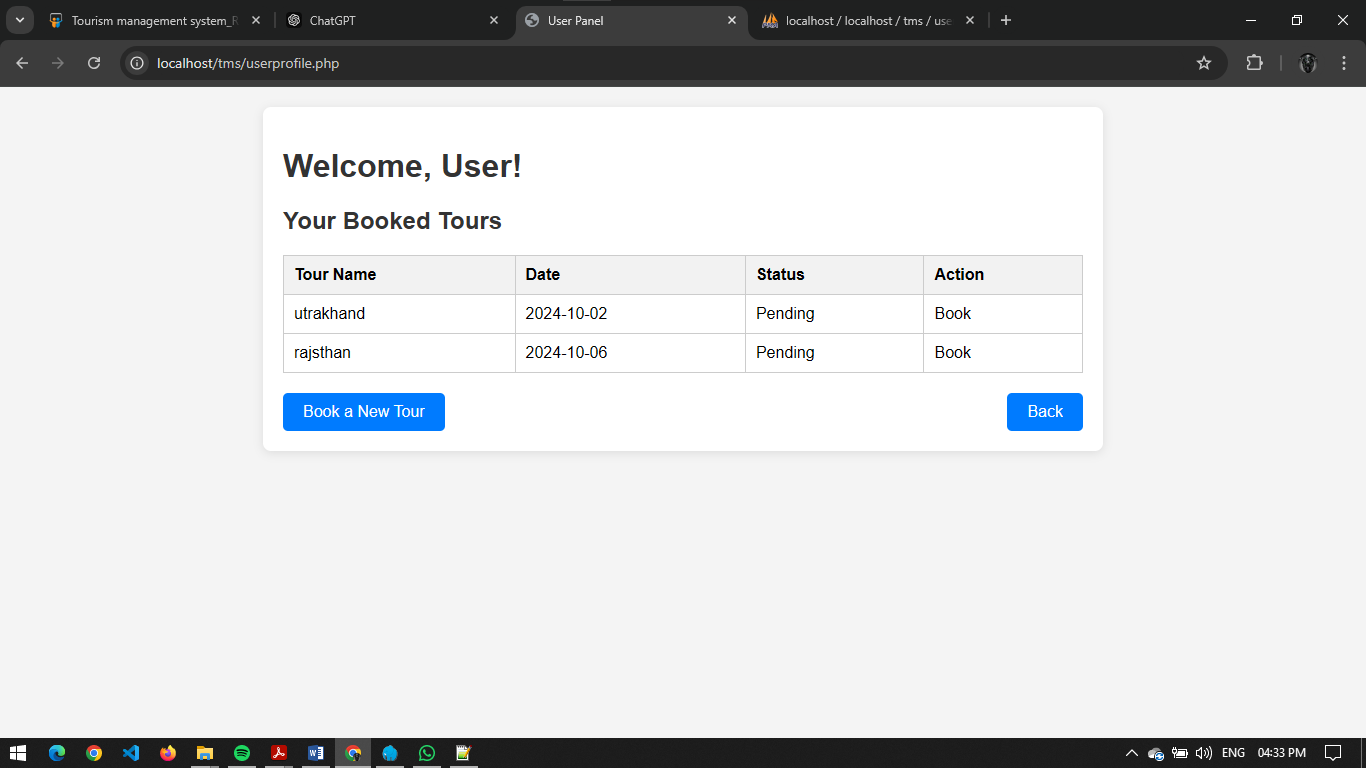
**User Login:**

****

**User Registration:**

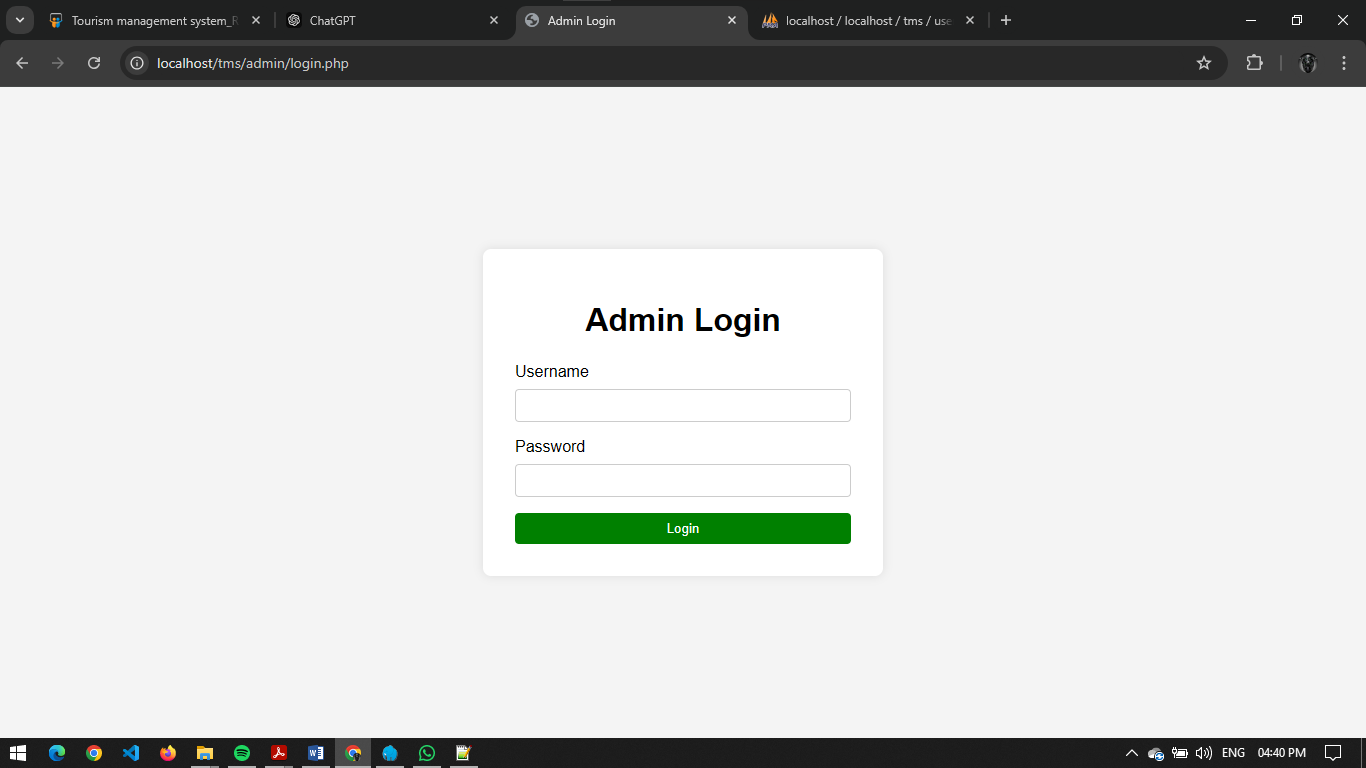
****

**User Profile:**

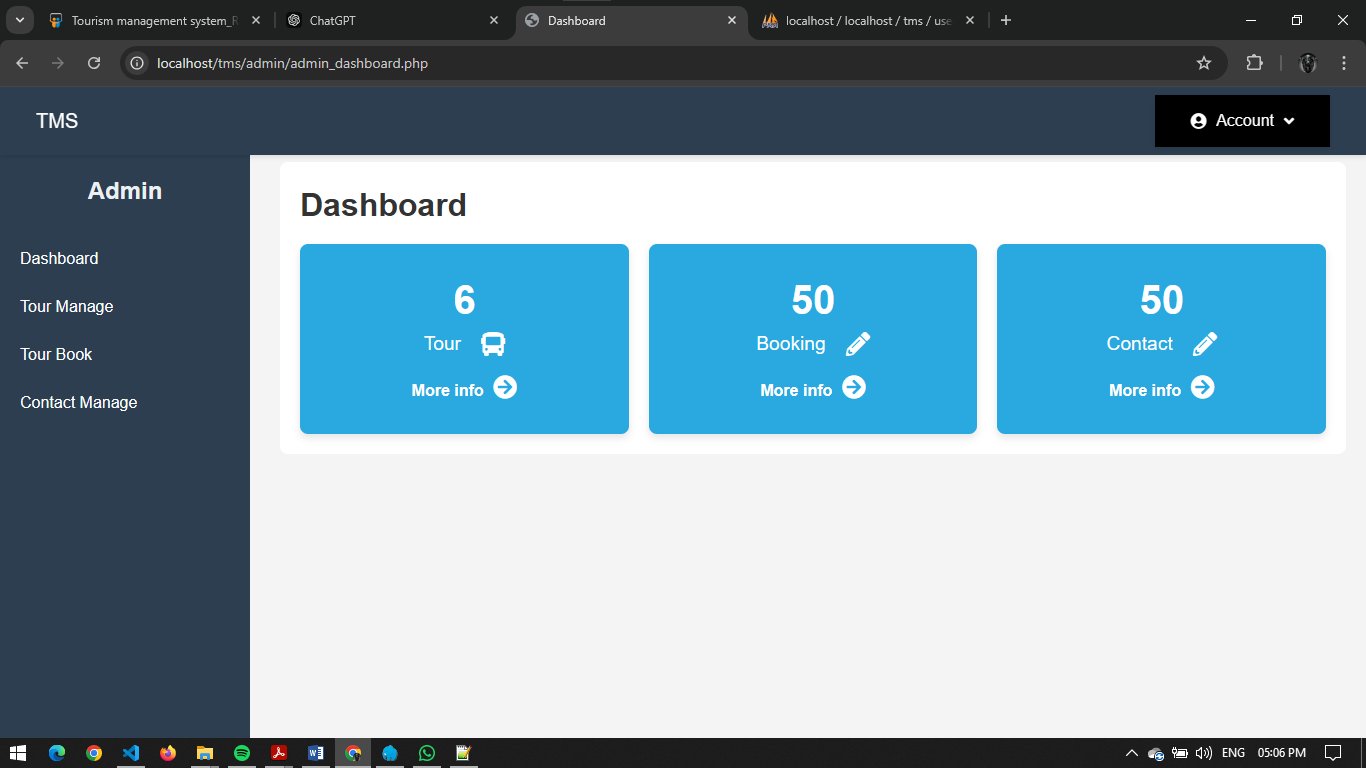
****

**Admin Panel:**

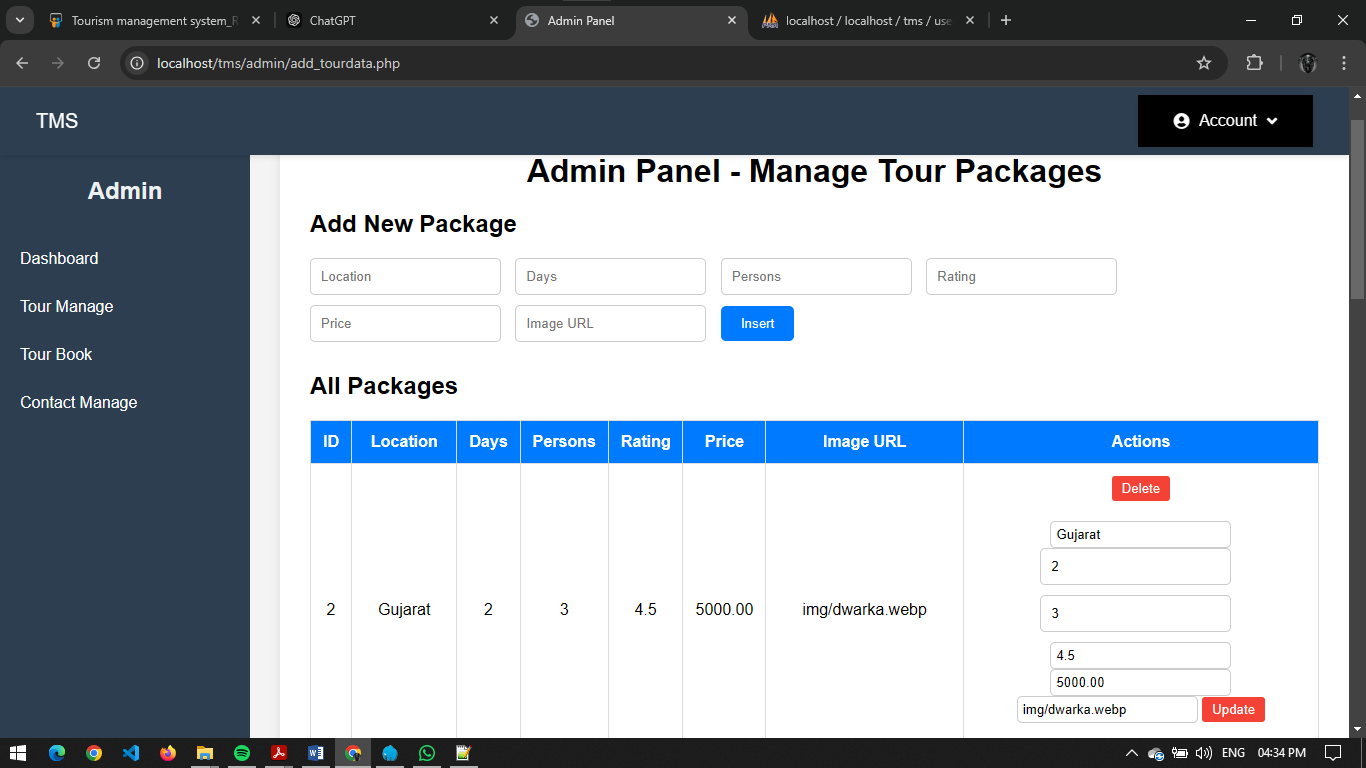
**Admin Login:**

****

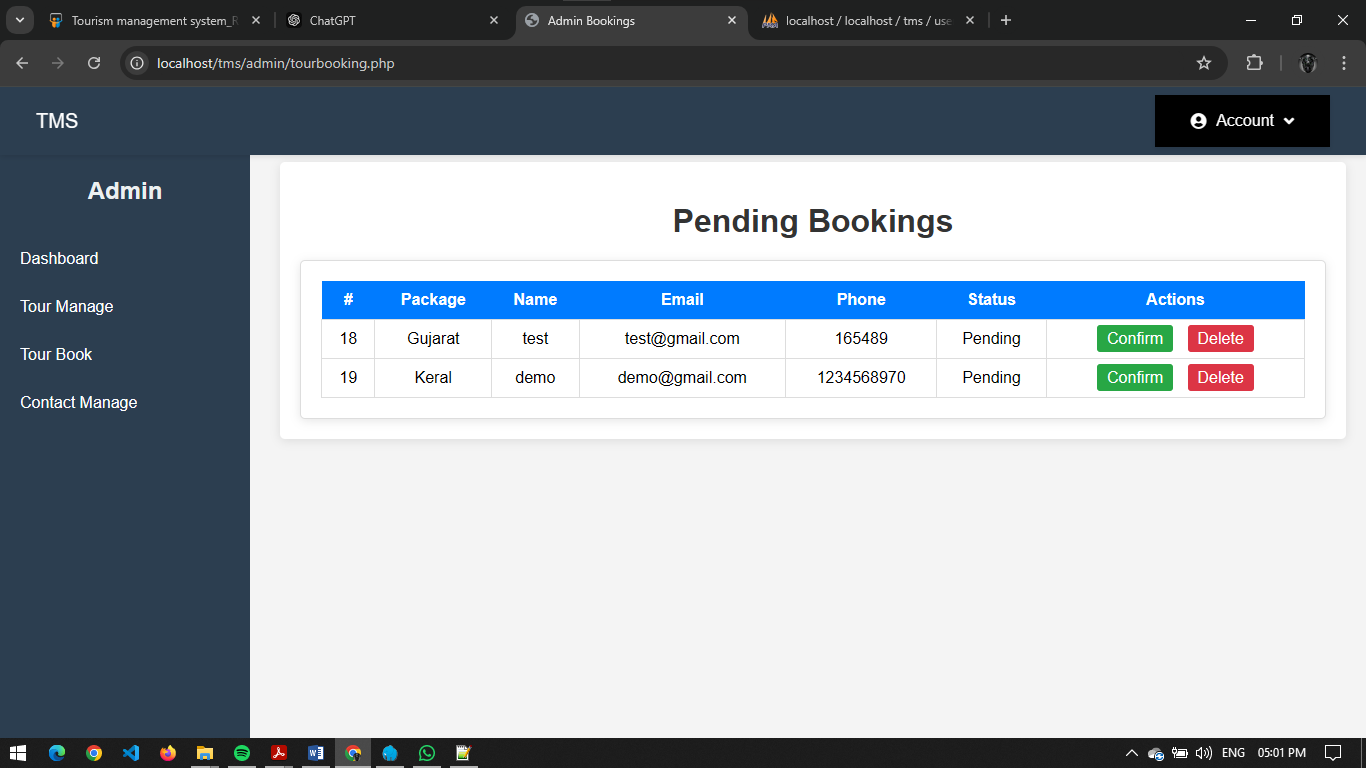
**Admin Dashboard:**

****

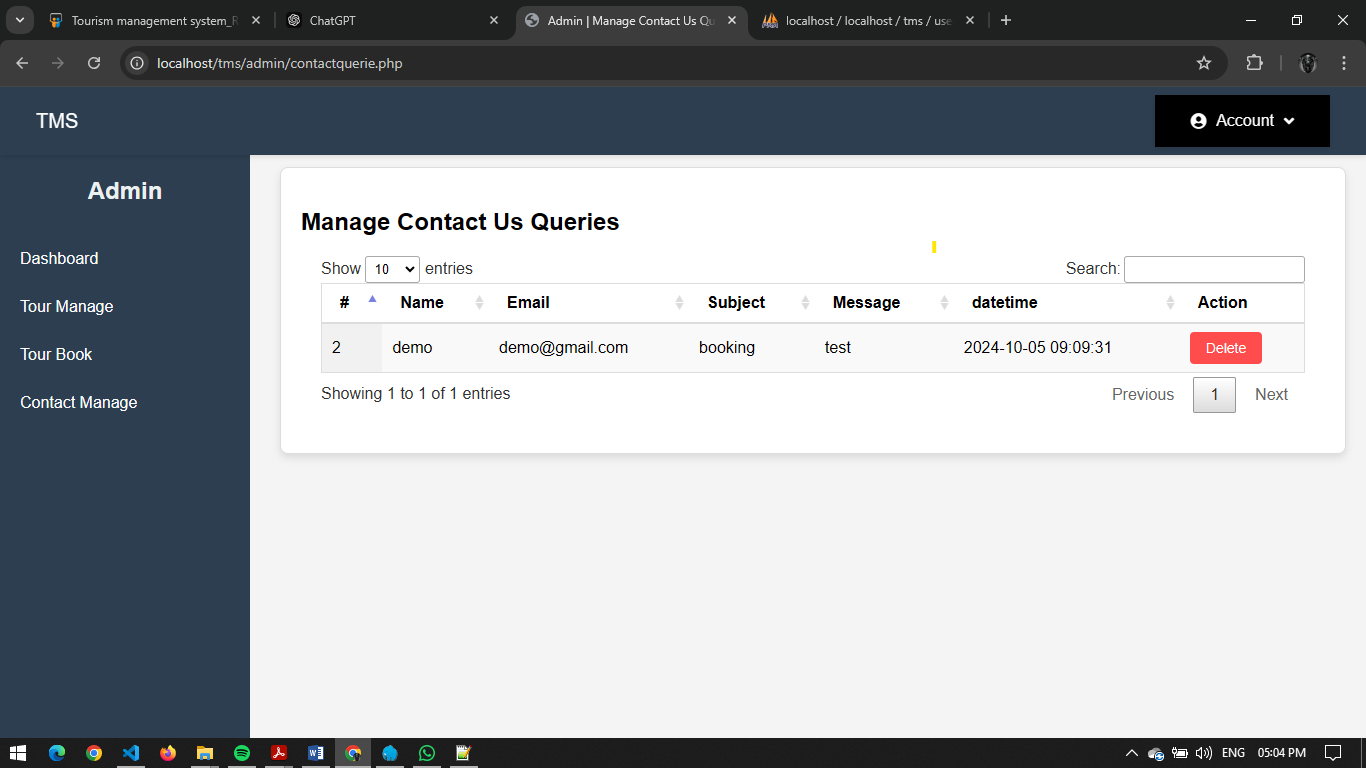
**Tour Manage:**

****

**Tour Book:**

****

**Contact Manage:**

****

**SYSTEM TESTING**

System testing is the phase before system implementation, where the system is thoroughly tested to ensure it is error-free and meets all necessary requirements. During this stage, test data is used to identify errors and implement corrections. The reports generated during testing are reviewed and approved by the users. The system is designed to be user-friendly, with features such as online help to assist users as needed.

**Test Plan:**

A **test plan** is a detailed document outlining the scope, strategy, and schedule of the testing process. It also identifies the test items and personnel responsible for different testing activities. The test plan for a project is critical in ensuring that all aspects of the system are thoroughly evaluated before it goes live.

**Major Testing Activities:**

The test plan covers several key activities:

* **Test units**: Identifying the individual components or units of the system to be tested.
* **Features to be tested**: Specific system features that need to be evaluated.
* **Testing approach**: Describes the methods and strategies that will be used during testing.
* **Test deliverables**: Outlines what documentation and results are to be produced during the testing process.
* **Schedule**: Specifies the timeline and deadlines for each stage of testing.
* **Personnel allocation**: Identifies the individuals responsible for carrying out the testing activities.

**Test Units:**

Testing is conducted at two levels: **Unit Testing** and **System Testing**.

**Unit Testing:**

Unit testing focuses on individual components or functions of the system to ensure they work as intended. The key units tested include:

* **Validating user requests**: Ensures that requests made by users are processed correctly.
* **Validating user input**: Checks that the system accepts and processes input data accurately.
* **Exception handling**: Tests how the system responds to errors or exceptions during operation.

**System Testing:**

System testing is conducted after unit testing to ensure that the integrated system functions as expected. Key aspects tested during system testing include:

* **Integration of programs**: Verifies that all programs and components are integrated correctly.
* **Full system functionality**: Ensures that the entire system operates as intended once all components are working together.

**Other Testing Strategies:**

**Alpha Testing:**

**Alpha testing** is conducted at the developer’s site, often with the customer present. The software is tested in a controlled environment, where the developer can monitor the user’s interaction with the system, identifying any issues or usability problems. Errors and other problems encountered during alpha testing are recorded for correction.

**Beta Testing:**

**Beta testing** takes place at the customer’s site and involves end-users using the software in a real-world environment. Unlike alpha testing, the developer is not present during beta testing. Users report any issues they encounter back to the

developer. This testing phase provides valuable feedback from actual users, which is used to make further improvements before the final product is released to the broader customer base.

**Test Deliverables:**

The testing process results in several important documents, which serve as evidence that the system has been thoroughly tested:

* **Unit Test Report**: A report documenting the results of unit testing for each component of the system.
* **Test Case Specification for System Testing**: A detailed document outlining the test cases used during system testing, which must be reviewed before testing begins.
* **System Testing Report**: A report summarizing the findings of system testing, including any issues found and corrective actions taken.
* **Error Report**: A log of any errors or bugs encountered during the testing process, along with their status (e.g., resolved or pending).

These documents are essential for ensuring transparency in the testing process and providing a record of the system's performance during testing.

**IMPLEMENTATION AND EVALUATION**

During the software-testing phase each module of software is thoroughly tested

for bugs and for accuracy of output. The system developed is very user-friendly and the detailed documentation is also given to the user as online help wherever necessary. The implementation phase normally ends with the formal test involving all the components.

The entire system was developed using the PHP, HTML, JavaScript, Personal Web Server, and MYSQL as back end. The HTML is used to design the web page. The Personal Web Server is used to understand the client’s request and to send response to them. The JAVASCRIPT are used for client-side validations so that the user can enter only appropriate input in the input fields. The MYSQL is the back end tool where the database resides.

Hence the design of the entire system is user-friendly and simple the implementation has been quite easy.