**Wonderlust Tours Documentation**

**Table of Contents**

1. Introduction

2. Project Setup

1. Prerequisites

2. Installation

3. Configuration

3. Project Structure

4. Database Schema

5. Application Routes

6. Templates

7. Styling

8. Application Features

9. Deployment

10. Contributing

11. License

12. Acknowledgements

**Introduction**

Wonderlust Tours is a web application for discovering and booking tours and travel packages. The application is built using Flask, a lightweight web framework in Python. This documentation provides a comprehensive guide on how to set up, use, and contribute to the Wonderlust Tours project.

**Project Setup**

**Prerequisites**

Before setting up the project, ensure you have the following installed:

- Python 3.8+

- pip (Python package installer)

- Virtual environment tool (e.g., venv or virtualenv)

- SQLite (for the database)

Installation

1. Clone the Repository:

git clone https://github.com/yourusername/WonderlustTours.git

cd Wonderlust Tours

2. Create a Virtual Environment:

python -m venv venv

source venv/bin/activate On Windows, use `venv\Scripts\activate

3. Install Dependencies:

pip install -r requirements.txt

Configuration

1. Environment Variables:

Create a `.env` file in the root directory and add the following variables:

FLASK\_APP=run.py

FLASK\_ENV=development

DATABASE\_URL=sqlite:///WonderlustTours.db

SECRET\_KEY=24

2. Initialize the Database:

flask db init

flask db migrate -m "Initial migration"

flask db upgrade

**Project Structure:** The project structure is as follows:



**Database Schema**

The application uses the following database schema:

1. User:

- `id` (Integer, Primary Key, Auto Increment)

- `created` (Timestamp, Not Null, Default Current Timestamp)

- `username` (Text, Not Null, Unique)

- `email` (Text, Not Null, Unique)

- `first\_name` (Text)

- `last\_name` (Text)

- `phone\_number` (Text)

- `password` (Text, Not Null)

2. Tour:

- `id` (Integer, Primary Key, Auto Increment)

- `title` (Text, Not Null)

- `description` (Text, Not Null)

- `price` (Real, Not Null)

- `location` (Text, Not Null)

- `image\_filename` (Text, Not Null)

- `created` (Timestamp, Not Null, Default Current Timestamp)

3. Booking:

- `id` (Integer, Primary Key, Auto Increment)

- `user\_id` (Integer, Not Null, Foreign Key to `users(id)`)

- `tour\_id` (Integer, Not Null, Foreign Key to `tours(id)`)

- `status` (Text, Not Null, Default 'pending')

- `title` (Text, Not Null)

- `location` (Text, Not Null)

- `price` (Text, Not Null)

- `username` (Text, Not Null)

- `email` (Text, Not Null)

- `first\_name` (Text)

- `last\_name` (Text)

- `date\_of\_booking` (Date)

- `created` (Timestamp, Not Null, Default Current Timestamp)

**Application Routes**

- Home Page: `/` - Renders the homepage.

- Tour Listings: `/tours` - Displays available tours.

- Book Tour: `/book/<int:tour\_id>` - Book a tour.

- User Account: `/account` - Displays user info and trips.

- Register: `/register` - Register a new user.

- Login: `/login` - User login.

- Logout: `/logout` - User logout.

- Tour Details: `/tour\_details/<int:tour\_id>` - Tour details.

- Admin Dashboard: `/admin/dashboard` - Admin dashboard.

- Add Tour: `/admin/add\_tour` - Add a new tour.

- Edit Tour: `/admin/tours/edit/<int:tour\_id>` - Edit an existing tour.

- Delete Tour: `/admin/tours/delete/<int:tour\_id>` - Delete a tour.

- Manage Bookings: `/admin/bookings` - Manage tour bookings.

- Confirm Booking: `/admin/confirm\_booking/<int:booking\_id>` - Confirm a booking.

- Deny Booking: `/admin/deny\_booking/<int:booking\_id>` - Deny a booking.

**Templates**

The application uses the following HTML templates located in the `templates/` directory:

- Base Template:

- `layout.html`: Common structure for other templates.

- Main Templates:

- `index.html`: Home page.

- `tours.html`: Tour listings page.

- `tour\_detail.html`: Tour details page.

- `book.html`: Booking page.

- `account.html`: User account management page.

- `login.html`: Login page.

- `register.html`: Registration page.

**- Admin Templates (in `templates/admin/`):**

- `create\_tours\_form.html`: Form to add new tours.

- `dashboard.html`: Admin dashboard.

- `main.html`: Main admin page.

- `navbar.html`: Navigation bar for admin.

- `sidebar.html`: Sidebar for admin navigation.

- `confirm\_tours.html`: Confirm tour bookings.

- `delete\_tours.html`: Delete tours.

- `deny\_tours.html`: Deny tour bookings.

- `edit\_tours.html`: Edit tour details.

**Styling**

The application uses Bootstrap for styling and also custom CSS styles in `static/css`.

- CSS Files (in `static/css/`):

- `account.css`: Styles for the user account page.

- `admin.css`: Styles for admin pages.

- `book.css`: Styles for the booking page.

- `index.css`: Styles for the home page.

- `register.css`: Styles for the registration page.

- `styles.css`: General styles.

- `tours.css`: Styles for the tour listings page.

**Scripting**

- JavaScript Files (in `static/js/`):

- `index.js`: Scripts for the home page.

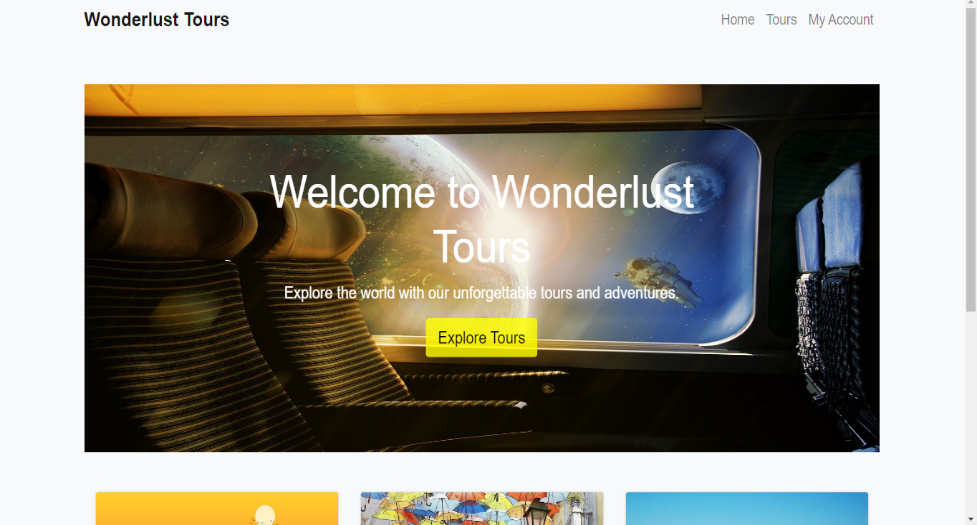
- `tours.js`: Scripts for the tour listings page.

**Images**

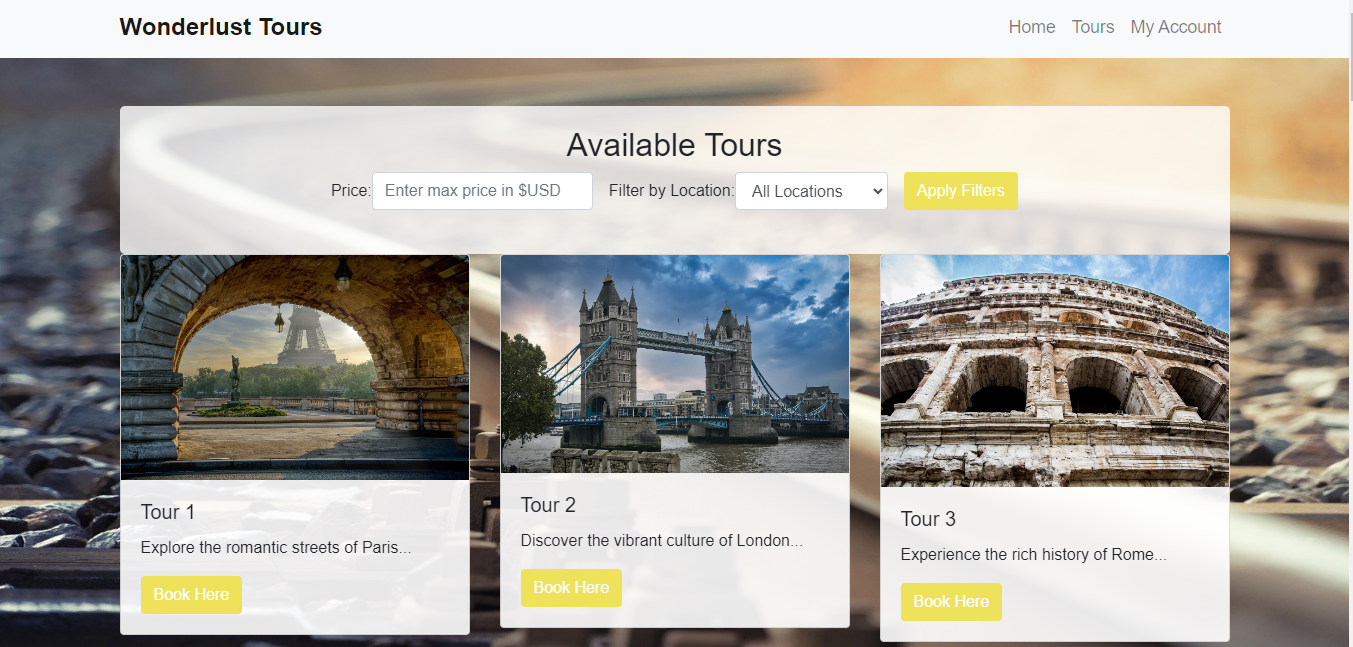
- **Images** (in `static/images/`): Contains multiple images used for styling and content.

**Application Features**

Wonderlust Tours offers the following key features

- Home Page: Welcomes users with an overview of services and a call-to-action button to explore tours.

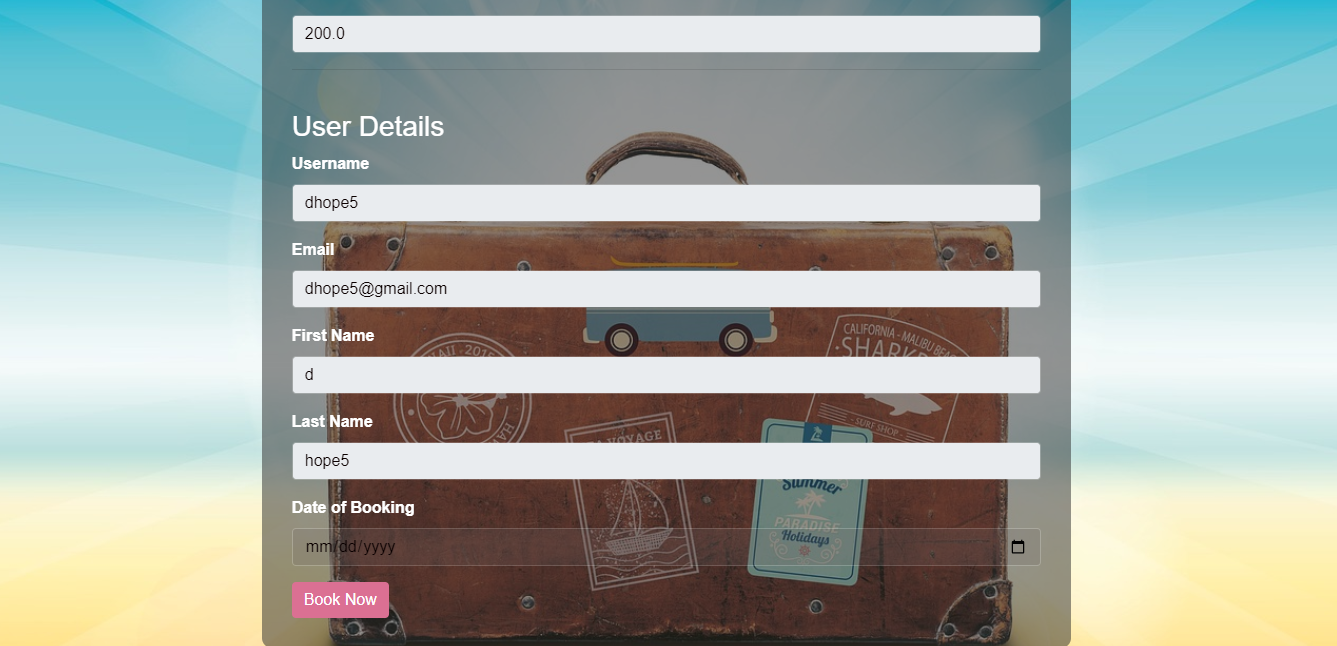
- Tour Listings: Displays all available tours in a grid layout with images, titles, brief descriptions, and links for more details.



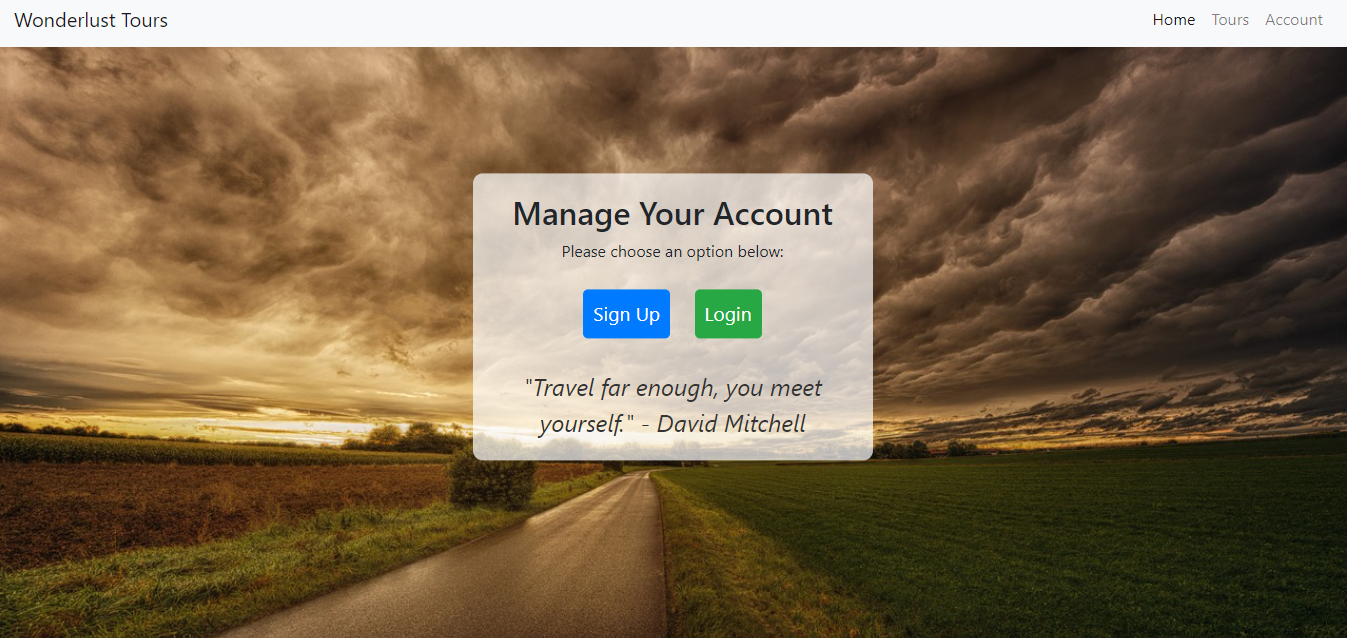
- Tour Details: Provides comprehensive information about a specific tour, including images, full description, pricing, and booking options.



- Booking: Allows users to book a tour by filling out a form with their details, which is then saved to the database.



- User Account Management: Displays user information and a list of their upcoming trips, including details of booked tours.



**Admin Features**

Wonderlust Tours includes the following admin functionalities:

- Dashboard: Provides an overview of the total tours, users, and bookings.

- Tour Management: Allows the admin to add, edit, and delete tours.

- Booking Management: Enables the admin to view all bookings, confirm or deny bookings, and manage booking statuses.

- Admin Account: Separate login for admin users to access the dashboard and management features.

**Responsive Design**

The application is fully responsive, ensuring a seamless experience across various devices, including desktops, tablets, and mobile phones.

**Deployment**

/\*

SOMEONE ADD THIS SEGMENT

\*/

**Acknowledgements**

- Flask: The web framework used.

- Bootstrap: The front-end framework used for styling.

- All contributors who have helped improve this project.