Detailed Design Document: TripForge

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed overview of the design and architecture of the TripForge, outlining its features, functionality, and technical specifications.

1.2 Scope

The website aims to offer users the users to book best suitable trips for themselves in their budget. They will be able to find the best Hotels, destinations. They can also book flight tickets for the trip. The users can also save the remainder and will get notification for their remainder.

1.3 Objectives

- Provide a user-friendly interface for booking Trips, hotels and flights.
- Implement robust security measures to protect user data.
- Ensure scalability to accommodate future growth.

2. System Overview

2.1 System Architecture

The system will follow a client-server architecture with a responsive web-based front end and a back-end server.

2.2 Key Features

- User registration and authentication
- Easy Trip booking and best nearby destinations.
- Ease in flight booking and remainder for events.
- Secured payment gateway.

2.3 User Roles

- Guest
- Registered User

2.4 Technologies Used

- Front-end: HTML, Tailwind CSS/CSS, JavaScript (React)
- Back-end: Node.js
- Database: MongoDB
- Authentication: JWT
- Additional tools/libraries as needed.

3. Database Design

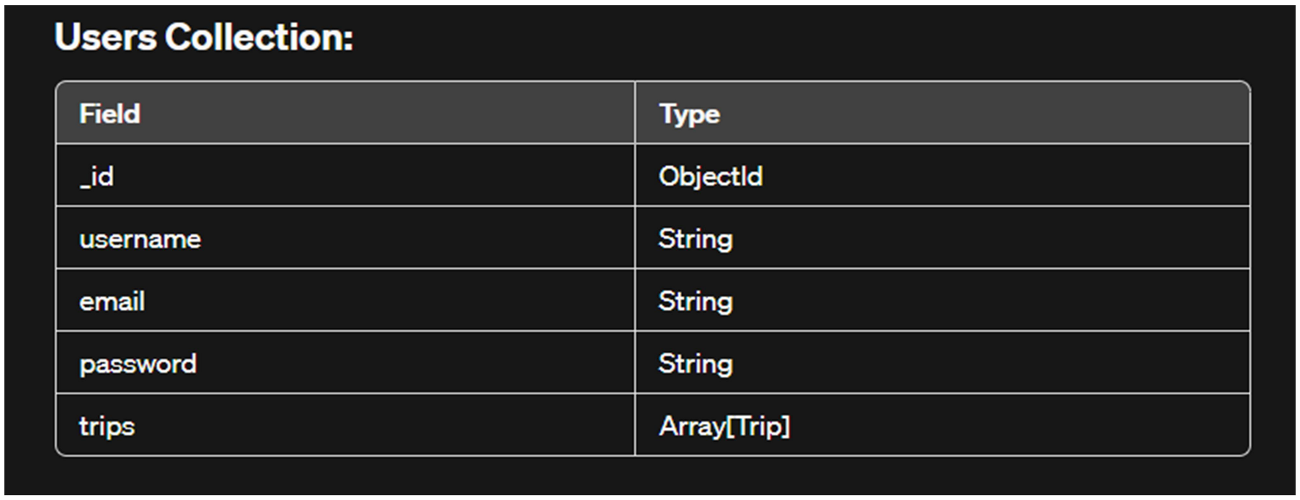
3.1 Entity-Relationship Diagram

- User Entity:
  - Username: User's username for identification.
  - Email: User's email address for communication.
  - Password: Securely stored password for authentication.

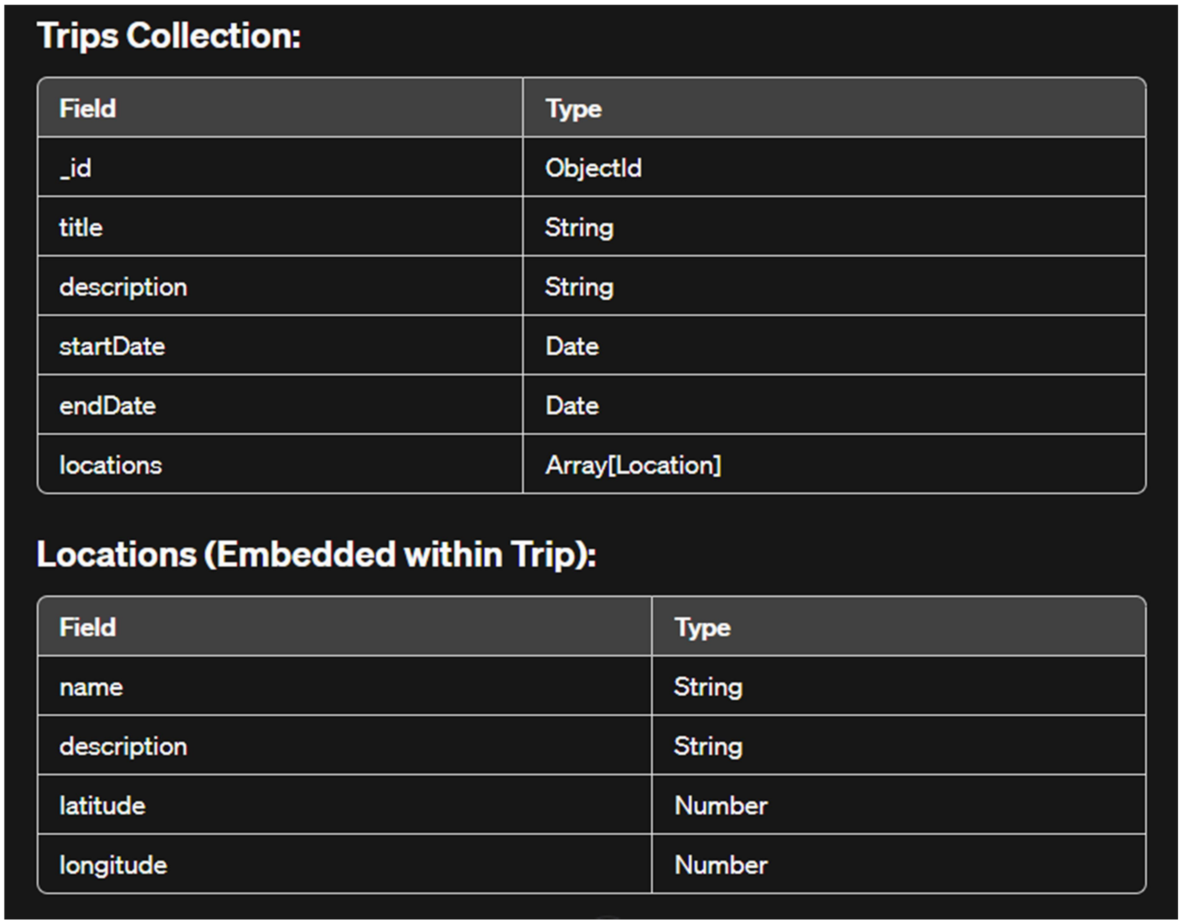
- Trips Entity:
  - Category: User’s trip preference (with family or with organization or with business family)
  - Budget : User’s budget for the trip.
  - TripType : No of people in the trip.
  - Destinations : All the destinations that the user will travel to during the trip.

- Flight Entity :
  - FlightBookingOption : Whether the user wants to book flight or not.
  - Departure Date : When the user will depart For the trip.
  - Return Date : When the user will return from the trip.

- Reviews Entity:
  - UserReviews : The reviews of users for the trip.


Users Collection:

Field	Type
_id	ObjectId
username	String
email	String
password	String
trips	Array[Trip]



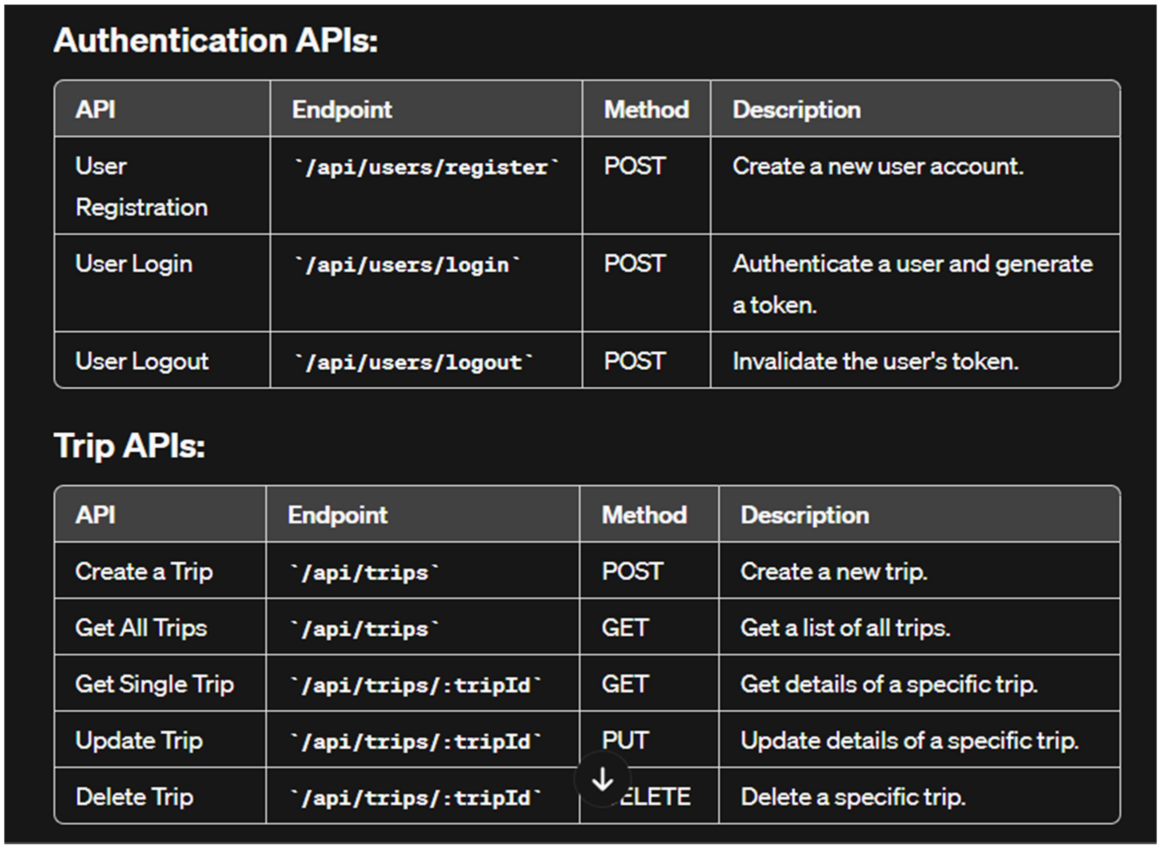
Trips Collection:

Field	Type
_id	ObjectId
title	String
description	String
startDate	Date
endDate	Date
locations	Array[Location]

Locations (Embedded within Trip):

Field	Type
name	String
description	String
latitude	Number
longitude	Number

4. APIs

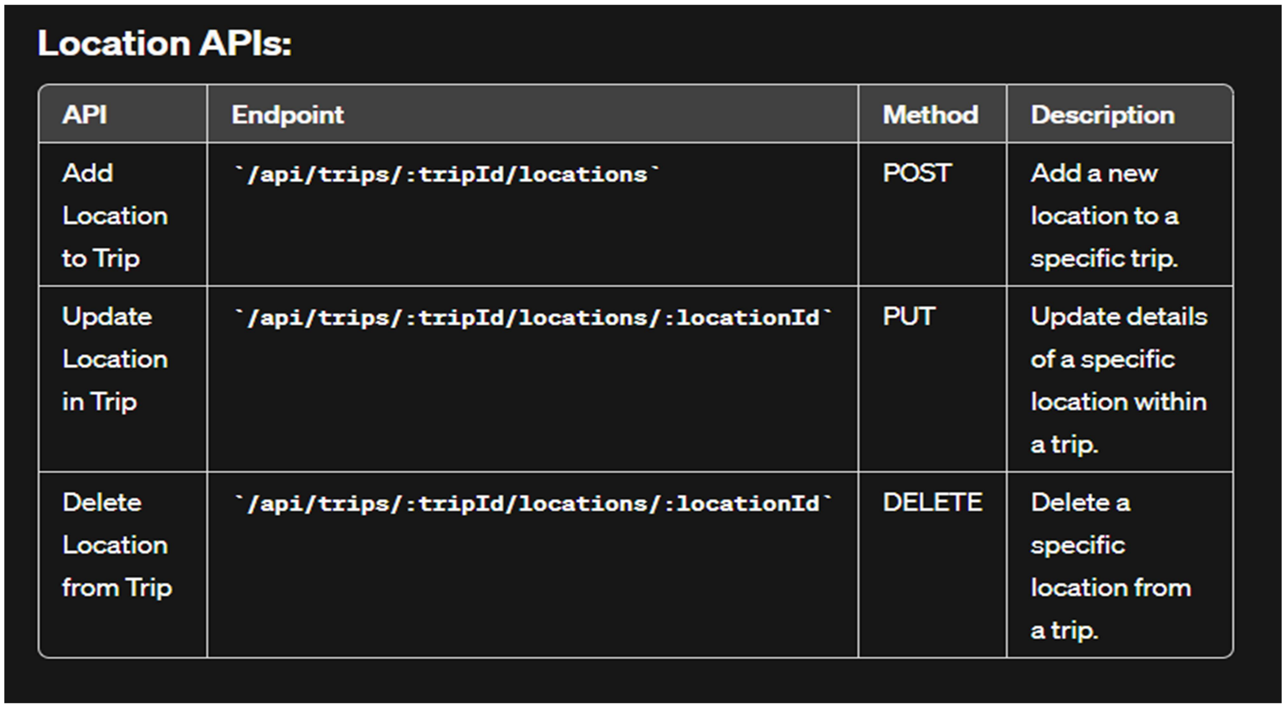


Authentication APIs:

API	Endpoint	Method	Description
User Registration	`/api/users/register`	POST	Create a new user account.
User Login	`/api/users/login`	POST	Authenticate a user and generate a token.
User Logout	`/api/users/logout`	POST	Invalidate the user's token.

Trip APIs:

API	Endpoint	Method	Description
Create a Trip	`/api/trips`	POST	Create a new trip.
Get All Trips	`/api/trips`	GET	Get a list of all trips.
Get Single Trip	`/api/trips/:tripId`	GET	Get details of a specific trip.
Update Trip	`/api/trips/:tripId`	PUT	Update details of a specific trip.
Delete Trip	`/api/trips/:tripId`	ELETE	Delete a specific trip.



Location APIs:

API	Endpoint	Method	Description
Add Location to Trip	`/api/trips/:tripId/locations`	POST	Add a new location to a specific trip.
Update Location in Trip	`/api/trips/:tripId/locations/:locationId`	PUT	Update details of a specific location within a trip.
Delete Location from Trip	`/api/trips/:tripId/locations/:locationId`	DELETE	Delete a specific location from a trip.

5. Deployment:

The TripForge Website Will be Deployed on Vercel.