Table of Contents

[1.Introduction 3](#_Toc154612186)

[2.Vision 3](#_Toc154612187)

[3.Scope 3](#_Toc154612188)

[4. System Features 3](#_Toc154612189)

[**a. Where to Fly:** 4](#_Toc154612190)

[**b. Packages:** 4](#_Toc154612191)

[**c. Check Bookings:** 4](#_Toc154612192)

[**d. Search Flight:** 4](#_Toc154612193)

[**e. Blood Bank:** 4](#_Toc154612194)

[**f. Technology Stack** 4](#_Toc154612195)

[**5.Non-Functional Features** 5](#_Toc154612196)

[**1. Performance** 5](#_Toc154612197)

[**2. Usability** 5](#_Toc154612198)

[**4. Reliability and Availability** 5](#_Toc154612199)

[**6. Business Rules** 5](#_Toc154612200)

[**7. Database Management System: MongoDB** 6](#_Toc154612201)

[**Scheme** 6](#_Toc154612202)

[**index.js** 6](#_Toc154612203)

[**Sequence diagram** 7](#_Toc154612204)

"Once you have tasted flight, you will forever walk the earth with your eyes turned skyward, for there you have been, and there you will always long to return." - Leonardo da Vinci

Title: Everest Air

# 1.Introduction

The "Everest Air" project aims to revolutionize the travel and tourism industry by offering a comprehensive online platform for booking flights, exploring destinations, and accessing exclusive travel packages. Everest Air envisions providing a seamless and user-centric experience to travelers, empowering them to efficiently plan and manage their journeys.

# 2.Vision

Everest Air aspires to become a leading online travel platform, recognized for its user-friendly interface, extensive flight offerings, customizable travel packages, and exceptional customer service. Our vision is to simplify the travel experience, making it more accessible, convenient, and enjoyable for users worldwide.

# 3.Scope

The scope of the Everest Air project encompasses the development of a dynamic web-based application that caters to various aspects of travel management. It includes but is not limited to:

*Flight Booking: Users can explore a wide range of flights, view details, and book tickets hassle-free.*

*Destination Exploration: Providing users with comprehensive information about destinations, attractions, and travel guides.*

*Package Customization:Offering diverse travel packages that cater to different preferences and budgets.*

*Booking Management:Allowing users to manage their bookings, make changes, and check flight details.*

*Flight Search:Implementing a robust search functionality for users to find specific flights based on their preferences.*

# 4. System Features

## **a. Where to Fly:**

Explain the functionality allowing users to explore destinations and flights available through Everest Air. Detail how users can search for destinations, view flight details, and make selections.

## **b. Packages:**

Describe the packages offered by Everest Air, including their features, pricing, and how users can browse and choose from these packages.

## **c. Check Bookings:**

Elaborate on the process of checking bookings. Describe how users can review their booked flights, manage their reservations, and make changes or cancellations if needed.

## **d. Search Flight:**

Detail the flight search functionality. Explain how users can search for specific flights based on criteria like date, destination, price range, etc.

## **e. Blood Bank:**

Mention the integration of a Blood Bank feature. Explain its purpose, functionality, and how it enhances the user experience.

## **f. Technology Stack**

Outline the technologies utilized in the development of the Everest Air website:

**Frontend:***Mention the frontend technologies like HTML, CSS, JavaScript, React.js,*

**Backend**:*Explain the backend technologies such as Node.js, Express.js, and specifically mention the use of MongoDB as the database management system.*

# **5.Non-Functional Features**

## **1. Performance**

* Response Time: Ensure quick response times for page loading and interaction within the website.
* Scalability: Design the system to handle increased user traffic during peak times without compromising performance.

## **2. Usability**

* User Interface (UI): Ensure a user-friendly interface that's intuitive and easy to navigate for users of all levels.
* Accessibility: Design the website to be accessible to users with disabilities, complying with accessibility standards.
* Multi-device Support: Ensure responsiveness across various devices, including desktops, tablets, and mobile phones.

## **4. Reliability and Availability**

* Fault Tolerance:Design the system to handle failures gracefully, ensuring minimal disruption to users.
* High Availability:Employ redundancy and failover mechanisms to maintain uninterrupted service availability.
* Backup and Recovery: Implement regular backups and reliable data recovery strategies to prevent data loss.

These non-functional features collectively contribute to the overall effectiveness, usability, security, and reliability of the Everest Air website, ensuring a seamless and secure experience for users while meeting industry standards and regulatory requirements.

# **6. Business Rules**

List and elaborate on the business rules governing the Everest Air website. These rules can include:

- User authentication and authorization processes

- Booking and payment rules

- Data privacy and security measures

- Handling customer queries and support

# **7. Database Management System: MongoDB**

Provide specific details about the use of MongoDB in the project. Explain how it's utilized for data storage, management, and retrieval within the Everest Air website.

## **Scheme**

import mongoose from "mongoose";

//attributes of the table/collection

const airApplicationSchema = new mongoose.Schema({

  firstName: String,

  lastName: String,

  passport: String,

  age: Number,

  gender: String,

  departureAirport: String,

  arrivalAirport: String,

  departureDate: Date,

  returnDate: Date,

  address:String,

  email: String,

  phone: String,

});

//basically this table/collection is called AirApplication in mongodb

const AirApplication = mongoose.model("AirApplication", airApplicationSchema);

export default AirApplication;

## **index.js**

import express from "express";

import mongoose from "mongoose";

//These are middleware

import cors from "cors";

import bodyParser from "body-parser";

import router from "./routes/airApplication.js";

const app = express();

mongoose

  .connect("mongodb+srv://salar:salar2001@cluster0.czepkzc.mongodb.net/", {

    useNewUrlParser: true,

    useUnifiedTopology: true,

  })

  .then(() => console.log("Connected to MongoDB"));

app.listen(5000);

//middleware --sequence of this very important

//this below one must be used first

app.use(cors());

//this below one must be used second

app.use(bodyParser.json({ extended: true }));

//this below one must be used third

app.use(bodyParser.urlencoded({ extended: true }));

app.use("/WhereToFly",router);

app.use("/CheckBookings",router);

### **Sequence diagram**



