

**A
PROJECT
ON
“CHEAPEST FLIGHT PRICE
TRACKER”**

Submitted to

Shiksha Mandal's
**G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)**
In the Partial Fulfillment of

B.Com. (Computer Application) Final Year

Submitted by

Atharva Pande
Atharva Kausadikar

Under the Guidance of

Pravin J. Yadao



Shiksha Mandal's
**G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)**
2021-2022

Shiksha Mandal's
**G. S. COLLEGE OF COMMERCE & ECONOMICS,
NAGPUR
(AUTONOMOUS)**

CERTIFICATE

(2021 - 2022)

**This is to certify that Mr. /Miss Atharva Pande and Atharva Kausadikar
has completed their project on the topic of "CHEAPEST FLIGHT PRICE TRACKER"
prescribed by G. S. College of Commerce & Economics, Nagpur (Autonomous) for
B.Com. (Computer Application) – Semester-VI.**

Date:

Place: Nagpur

Pravin J. Yadao

Project Guide

External Examiner

Internal Examiner

ACKNOWLEDGEMENT

We take this opportunity to express our deep gratitude and whole hearted thanks to project guide Prof. Pravin Yadao, Coordinator for his guidance throughout this work. We are very much thankful to him for his constant encouragement, support and kindness.

We are also grateful to our teachers Prof. Rahul Tiwari, Prof. Sushma Gawande, Prof. Preeti Rangari, Prof. Prajka Deshpande and Prof. Haresh Naringe for their encouragement, help and support from time to time.

We also wish to express our sincere thanks to Principal Dr. N. Y. Khandait for providing us wide range of opportunities, facilities and inspiration to gather professional knowledge and material without which this project could not have been completed.

Atharva Pande

Atharva Kausadikar

Date:

Place: Nagpur

DECLARATION

We **Atharva Pande & Atharva Kausadikar** hereby honestly declare that the work entitled "CHEAPEST FLIGHT PRICE TRACKER" submitted by us at G. S. College of Commerce & Economics, Nagpur (Autonomous) in partial fulfillment of requirement for the award of B.Com. (Computer Application) degree by Rashtrasant Tukadoji Maharaj, Nagpur University, Nagpur has not been submitted elsewhere for the award of any degree, during the academic session 2021-2022.

The project has been developed and completed by us independently under the supervision of the subject teacher and project guide.

Student Name & Signature

Date:

Place: Nagpur

INDEX

Sr. No.	Particulars	Page No.	Remark	Signature
1.	INTRODUCTION	1-2		
2.	OBJECTIVES	3-4		
3.	PRELIMINARY SYSTEM ANALYSIS 3.1. Preliminary Investigation 3.2. Present System in use 3.3. Flaws of New System 3.4. Need of New System 3.5. Feasibility Study	5-10		
4.	PROJECT CATEGORY	11-13		
5.	SOFTWARE AND HARDWARE REQUIREMENT SPECIFICATIONS	14-15		
6.	DETAILED SYSTEM ANALYSIS 6.1. Data Flow Diagram 6.2. ER Diagram 6.3. Data Tables	16-19		
7.	SYSTEM DESIGN 7.1. Form Design 7.2. Source Code 7.3. Input Output Screen	20-64		
8.	TESTING AND VALIDATION	65-67		
9.	SYSTEM SECURITIES MEASURES	68-69		
10.	IMPLEMENTATION, EVALUATION AND MAINTENANCE	70-72		
11.	FUTURE SCOPE OF PROJECT	73-74		
12.	CONCLUSION	75-76		
13.	BIBLIOGRAPHY	77-79		
14.	Approved Copy of Synopsis	80-82		

INTRODUCTION

Introduction

Our project Title is “Cheapest Flight Price Tracker” as the name suggests it helps to find the cheapest flight from one Destination to another destination. It is helpful for the customer to find out their journey at an affordable price. Nowadays, each and every person or specifically the young generation is tech-savvy. Almost everyone has a smartphone, pc, or laptop. They are user friendly with the internet as well as online application and software. So here is our website to provide a platform to find their flight/Aeroplane Journey Affordably.

The website saves your time from googling and helps in finding affordable flights by just a click. People/Customers just have to enter their journey details, that is the journey date and destination and rest all the details will be fetched. It contains information on schedules and fares and contains a database of reservations (or passenger name records) and of tickets issued (if applicable).

Price Alerts track flight prices in real-time and notify you when the price of your flight changes - that can be up or down! Flight prices, like it or not, tend to change very quickly with time. When you set up a Price Alert, whenever the price of your flight goes up, or down, you get a notification. This helps you find the ideal date and time when to book your flight and get the best price. Flight Price Tracker not only saves your money but also time and gets you the best deal according to your requirements.

OBJECTIVE

Objective

1)**To track Flight Price** - The website tracks the best flight from one location to another location by just a click saving the time of the customer. There is no need for the customer to check, by going to the airline office or contacting any broker.

2)**User friendly**- Website is user-friendly so that anyone can easily access it through pc, mobile, tablets etc. The user interface is also simple (to the point) so that any person can understand it without any hesitation.

3)**To provide Affordable service** - The website is going to provide very affordable and Economical service to the user/customer. The people of every class can book their flight. Basically, it is Budget-friendly.

4)**To get an intuition of future flight prices** – The website helps to understand the future price of any flight. Also, it provides a tentative pattern when the fare of any flight increases or decreases.

5)**Save time and Money** – It saves the most valuable thing of everyone, which is time. Users/Customers will be able to track their flight at anytime from anywhere as it is available 24/7. It saves money as there is no need for the users to visit any office or to pay a charge regarding any information.

PRELIMINARY SYSTEM ANALYSIS

Preliminary System Analysis

1)Preliminary Investigation: Preliminary Investigation basically refers to the collection of information that guides the management of an organization to evaluate the merits and demerits of the project request and make an informed judgment about the feasibility of the proposed system.The purpose of the preliminary investigation is to determine whether the problem or deficiency in the current system really exists. This sort of investigation provides us with a thorough picture of the kind of software and hardware requirements which are most feasible for the system, plus the environment in which the entire project has to be installed and made operational.

The preliminary-investigation phase sets the stage for gathering information about the current problem and the existing information system. This information is then used in studying the feasibility of possible information systems solutions.

Preliminary Investigation of the project:

The Preliminary Investigation took about five days. During this period, we overviewed a few research papers and many articles. We started with a research paper by [O. Etzioni, R. Tuchinda, C. A. Knoblock, and A. Yates. To buy or not to buy: mining airfare

data to minimize ticket purchase price]. Through this, we got an insight into how flight prices change according to region, dates, occasion, and time interval. We also overviewed [Groves and Gini, 2011. A Regression Model For Predicting Optimal Purchase Timing For Airline Tickets]. This paper presents a model for computing expected future prices and reasoning about the risk of price changes. By comparing models with different target properties, buyers can determine the likely cost of their preferences.

2)Present System in use

The present system in use is provided with lots of services and facilities. The system can be maintained as per user requirements and changes or updates can be made from time to time.

Our system provides flight fare data to the customer using various API to fetch the lowest price and the customer can afford it easily.

3)Flaws of New System

1) Lack Of Security- There are a lot of sources available for booking flight tickets but some of them are insecure and data of users may be leaked.

2) Difficult to understand UI- User cannot understand the user interface of the booking websites available in market. Good user interface necessary to attract customer.

3) Tracking- In the present system they just provide fix price but they may be expensive which cannot be affordable by customer.

4) Cost-Air travel is considered to be the most expensive means of transportation. As there are many websites available having different prices so user have to check each and every website and it is time consuming as well. That's why air travel is so expensive that it gets beyond ordinary people's grasp.

4)Need of New System

The trends of the prices are highly sensitive to the route, month of departure, day of departure, time of departure, whether the day of departure is a holiday, and airline carrier. Highly competitive routes like most business routes (tier 1 to tier 1 city like Mumbai and Delhi) had a non-decreasing trend where prices increased as days to departure decreased, however other routes (tier 1 to tier 2 cities like Delhi- Guwahati) had a specific time frame where the prices are minimum. Moreover, the data also uncovered two basic categories of airline carriers operating in India – the economical group and the luxurious group, and in most cases, the minimum priced flight was a member of the economical group. The data also validated the fact that there are certain time periods of the day where the prices are expected to be maximum therefore updation is necessary in the market.

Another problem here is that people who fly regularly always need to perform various comparisons constantly. This demands time and

often they pay a greater price. There is no need for the customers to pay a greater price if they can get the ticket at a cheaper price and at the same time they need not spend too much of their time while searching for the rightly priced flight ticket.

5)Feasibility Study

A feasibility study is an assessment that determines the likelihood of a proposed project being successful, such as a new product line or technical system. The study analyses the project's relevant factors, such as technical, economic and legal considerations, to assess whether the project is worth an investment. The study can also identify potential issues and problems that could arise from pursuing the project.

Feasibility studies also help companies with new business development, including determining how it will operate, potential obstacles, competition, market analysis and the amount and source of financing needed to grow the business. They can help develop marketing strategies to convince investors and banks that investing in a particular project or business is a wise choice.

1.Technical Feasibility-Technical feasibility is the formal process of assessing whether it is technically possible to manufacture a product or service. Before launching a new offering or taking up a client project, it is essential to plan and prepare for every step of the operation. Technical feasibility helps determine the efficacy of

the proposed plan by analyzing the process, including tools, technology, material, labor and logistics.

2. Economic Feasibility-This assessment typically involves a cost/benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3. Legal Feasibility-This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

4. Operational Feasibility-This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

PROJECT CATEGORY

Project Category

“Cheapest flight price tracker” is a project which is developed using web designing languages that are HTML, CSS, JavaScript, Python, Php the websites provide entire structure to track affordable flight fare and provide notification of selected flight by customer

HTML-

Used for designing the front-end of the website. HTML stands for Hypertext Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within a tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

CSS-

Used for designing the front-end of the website. CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

Python-

Used in the Backend for alert code and it is hosted on PythonAnywhere.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

JavaScript-

Used in the Backend to fetch flight data from the API. JavaScript (JS) is a scripting language, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code. JavaScript is an interpreted language. Thus, it doesn't need to be compiled. JavaScript renders web pages in an interactive and dynamic fashion. This allowing the pages to react to events, exhibit special effects, accept variable text, validate data, create cookies, detect a user's browser, etc

PHP-

Used for User Authentication. "Hypertext Preprocessor," a recursive acronym. PHP is a scripting language web developers use to create dynamic websites. It is also a general-purpose language that you can use to make lots of projects, including Graphical User Interfaces (GUIs). When a website visitor accesses a PHP page, the web server processes, or "parses," the PHP code, which can output HTML to the webpage.

Firestore-

Firestore database is used for user authentication and to store users' search query.

**SOFTWARE AND HARDWARE
REQUIREMENT SPECIFICATIONS**

SOFTWARE AND HARDWARE

A computer system is divided into two categories: Hardware and Software. Hardware refers to the physical and visible components of the system such as a monitor, CPU, keyboard and mouse. Software, on the other hand, refers to a set of instructions which enable the hardware to perform a specific set of tasks.

Software and Hardware Requirement

Hardware Requirements-

RAM - 4GB

Processor - 1.6 GHz or faster processor.

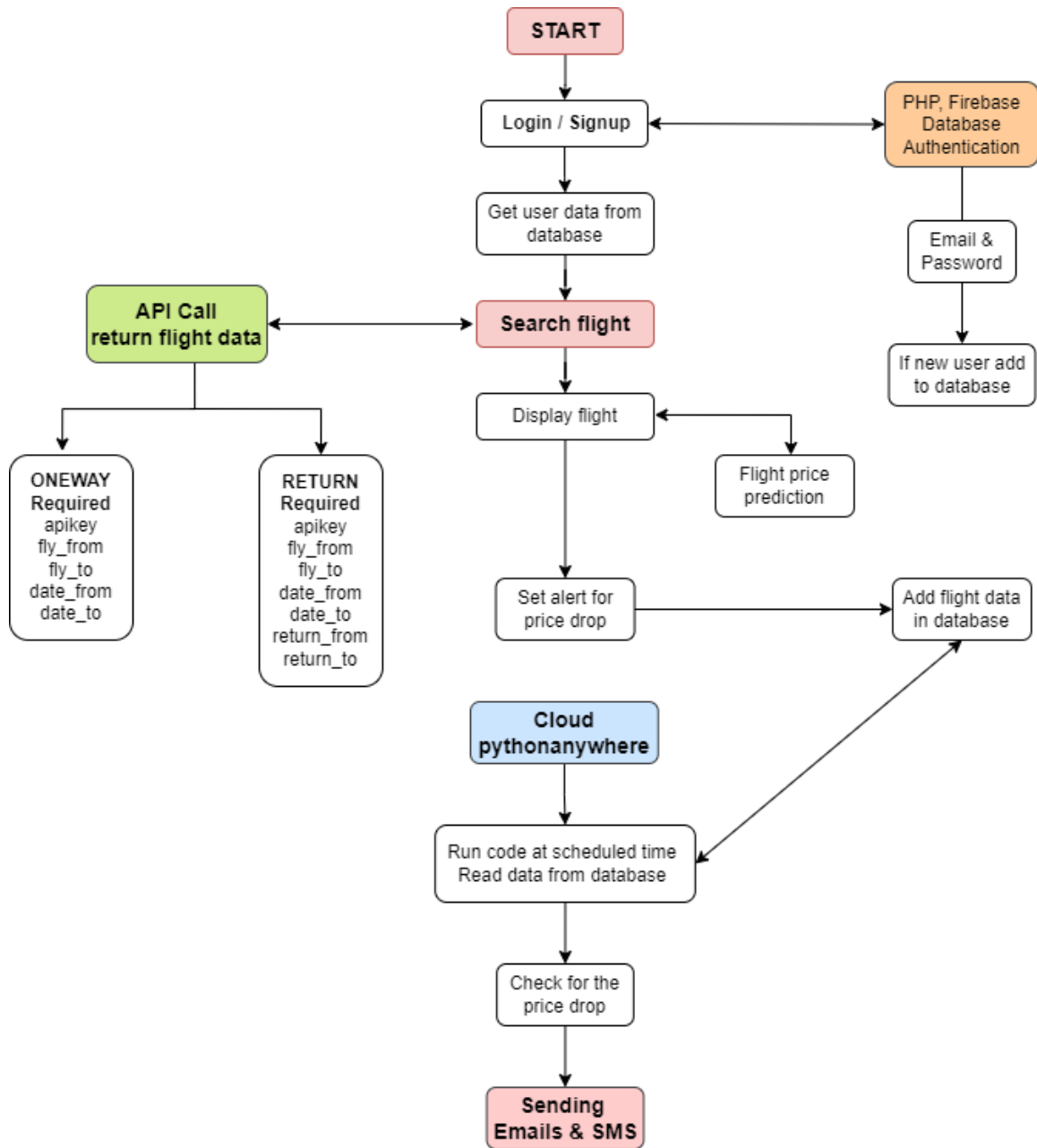
Operating System - Windows 8 or later

Any Browser (supports JavaScript) like Internet Explorer, Chrome

Currently, Flight Price Tracker only supports windows. To run the application, the computer should have decent RAM and a good CPU so that it does not lag and other applications can run as well. The application does not need any specific hardware component other than the absolute basic ones. The application is based on JavaScript, Python. Hence, it needs a JavaScript-compatible browser to run. The application depends on the internet to retrieve data and display it.

DETAILED SYSTEM ANALYSIS

Data Flow Diagram



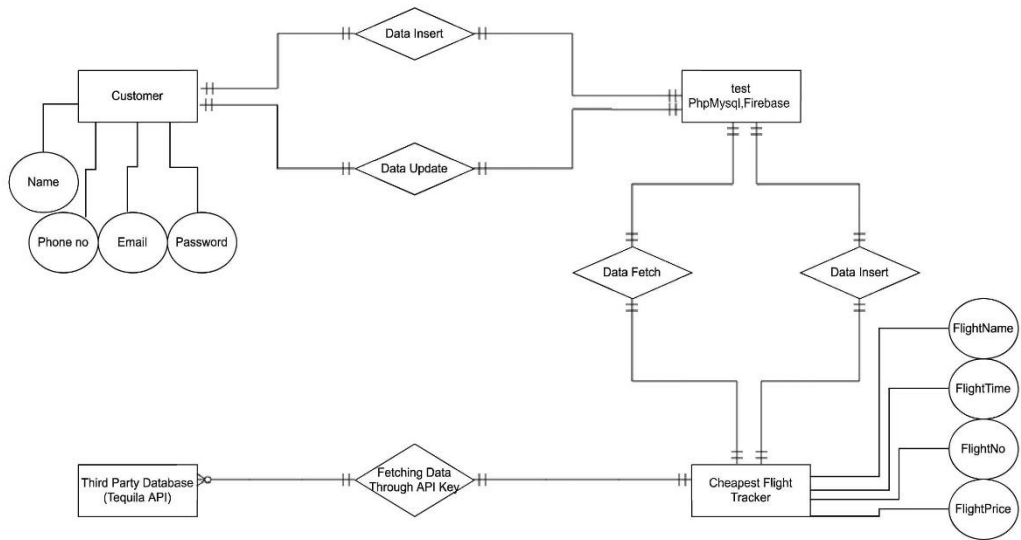
Once the user starts the application, he can choose to sign up/log in or simply check the prices. If the user decides not to log in, he can just enter the required details about the flight.

After this, the application will make some API calls and display the results on the screen. Since the user was not signed in, no price alerts can be set. If the user is a new customer and decides to sign up then all his credentials will be taken as input and authentication will be done through php and firebase. In the next step, a database will be created for new users. After this, the user will be taken to the home screen. If it's an existing customer then his credentials will be verified with the ones in the database.

After a user logs in he will be taken to the home screen where he can just enter the required details about the flight. After this, the application will make some API calls and display the results on the screen.

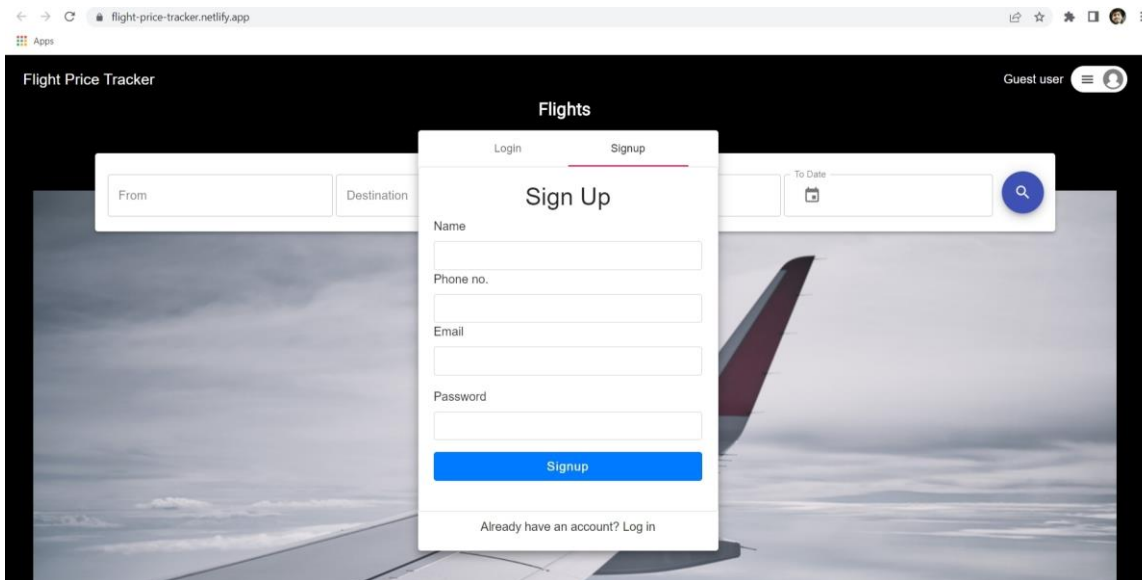
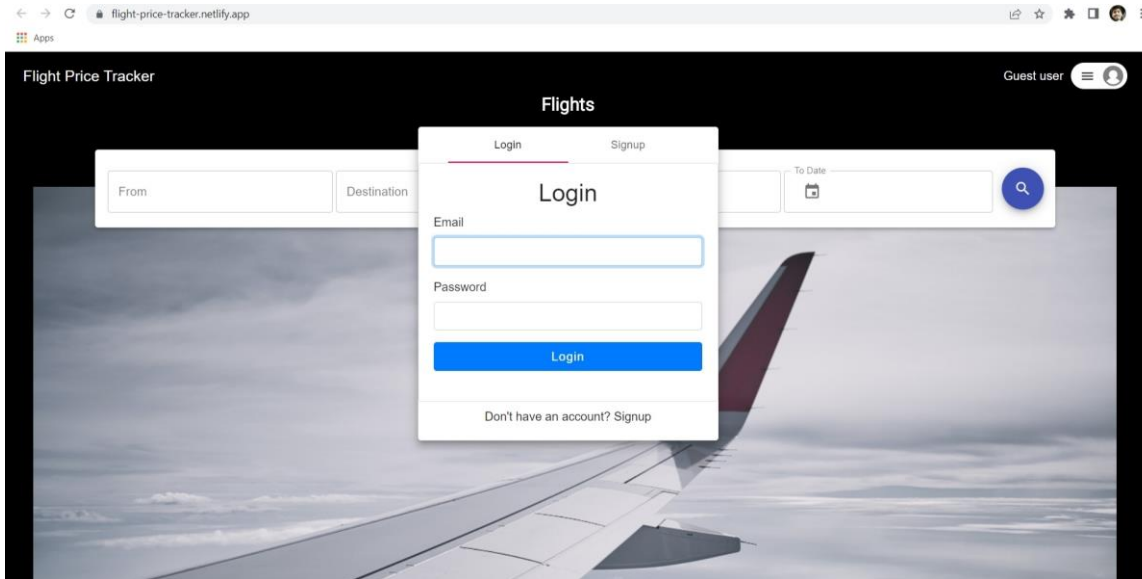
Since the user was signed in, a price alert can be set. A python code(hosted on PythonAnywhere) runs every day and reads data from php and firestore and checks for the price variation if the price either decreases or increases an email will be sent to the user mentioning the flight price details.

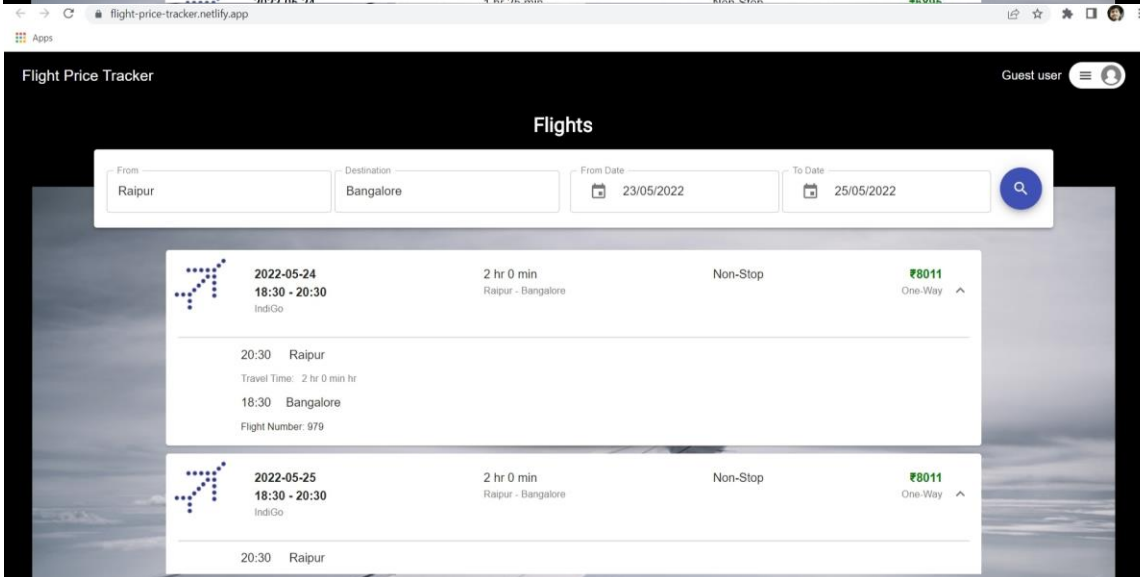
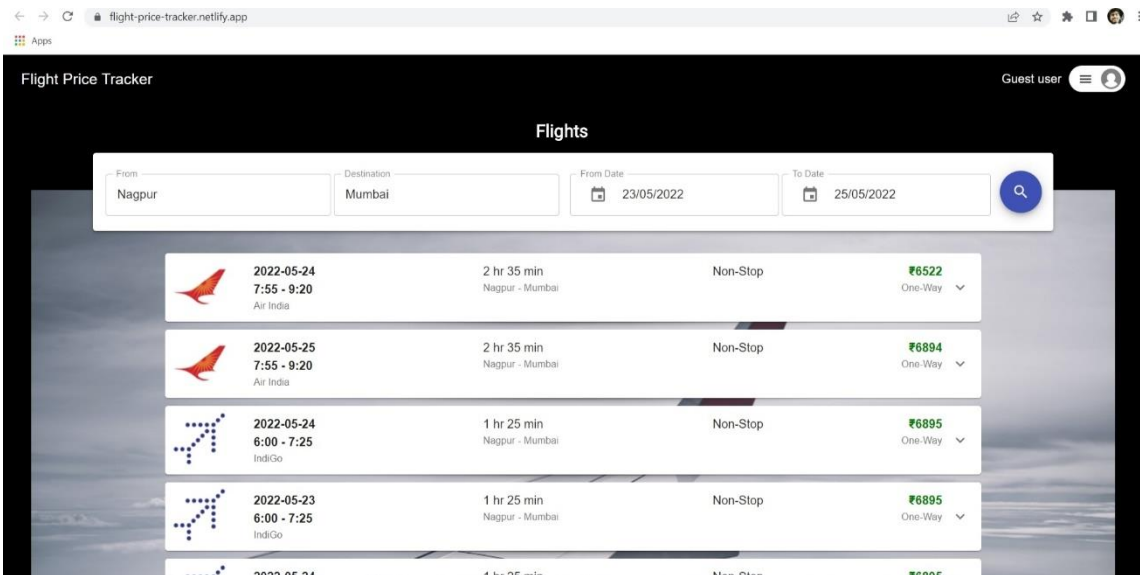
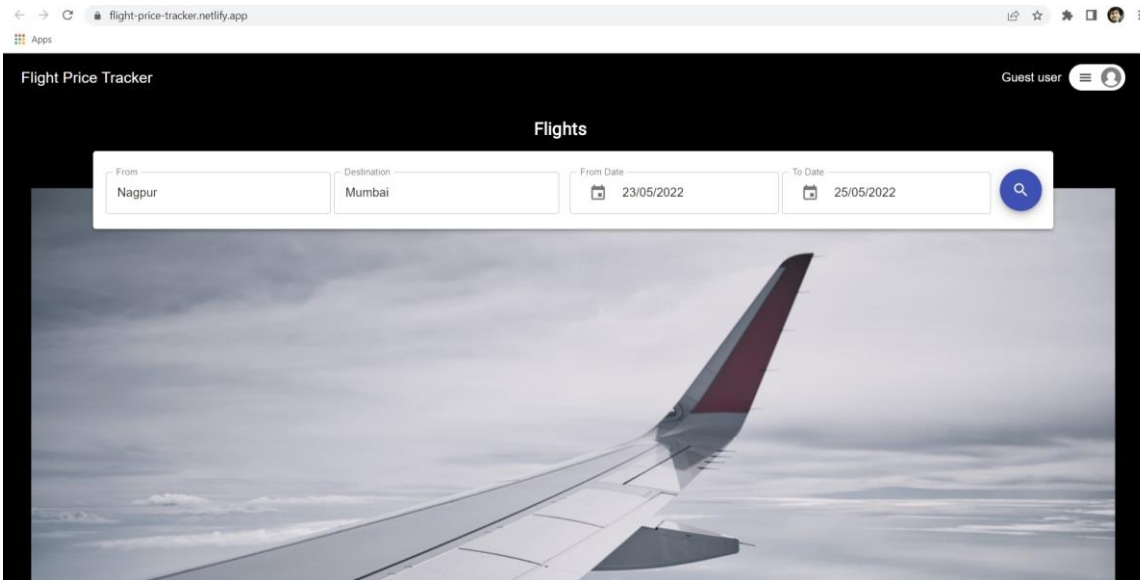
Entity Relationship Diagram



SYSTEM DESIGN

Form Design





DataTables

SELECT * FROM `user_form`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	name	password	email	user_type
<input type="checkbox"/> Edit Copy Delete	1	kanha	c20ad4d76fe97759aa27a0c99b6710	kanha@gmail.com	user
<input type="checkbox"/> Edit Copy Delete	2	himanshu	202cb962ac59075b964b07152d234b70	h@gmail.com	admin
<input type="checkbox"/> Edit Copy Delete	3	Charu	81dc9bdb52d04dc20036dbd8313ed055	charu@gmail.com	user
<input type="checkbox"/> Edit Copy Delete	4	sanket	81c6fa745434534c4a9633ad8c08506a	sanketmaive@gmail.com	admin
<input type="checkbox"/> Edit Copy Delete	5	Chandan Sharma	b18f2dfcfc1c94c17e43419b7e8f788b	chandan@gmail.com	admin
<input type="checkbox"/> Edit Copy Delete	6	Pande	62c428533830d84fd8bc77bf402512fc	pande@gmail.com	user
<input type="checkbox"/> Edit Copy Delete	7	ap	62c428533830d84fd8bc77bf402512fc	ap@gmail.com	user
<input type="checkbox"/> Edit Copy Delete	8	Pande	9a3856de24797a250024edd5847b5de0	pande@gmail.com	user
<input type="checkbox"/> Edit Copy Delete	9	ram	4641999a7679fcaef2df0e26d11e3c72	ram@gmail.com	user
<input type="checkbox"/> Edit Copy Delete	10	yk	529d2cd920682274fce648af585d016a	yk@gmail.com	user

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

console.firebase.google.com/project/flightpricetrackerpde/firestore/data/~2Fusers~2Fatharvapande200@gmail.com

Apps

flightpricetrackerPDE - Cloud Firestore

Go to docs

users > atharvapande200...

flightpricetrackerpde users atharvapande200@gmail.com

+ Start collection + Add document + Start collection

users > atharvapande200@gmail.com > + Add field

atharvapande34@gmail.com
 bhargavigawande95@gmail.com
 yash258@gmail.com
 yashcgh@gmail.com
 yashchaudhari1090@gmail.com
 yashchaudhar1109@gmail.com
 yyy@gmail.com

contact: "9359062360"
 email: "atharvapande200@gmail.com"
 name: "Atharva Pande"

query1
 city1: "NAG"
 city2: "BOM"
 date1: "2022-05-24T09:20:00.000Z"
 date2: "2022-05-24T07:55:00.000Z" (string)
 price: 6522

query10
 city1: "DEL"
 city2: "BOM"

Source Code

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+JP&display=swap"
      rel="stylesheet">

    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

  </body>
</html>
```

Login.js

```
// import React from 'react'
import React,{useRef} from 'react'
import {useEffect} from 'react'
import styled from 'styled-components'
import { makeStyles, withStyles } from '@material-ui/core/styles';
import { Form,Button,Card} from 'react-bootstrap'
import '../App'
```

```
const useStyles = makeStyles((theme) => ({
  card:{
    width:'500px'
  }
}));
```

```
function Login(props){
  const{
    email,
    setEmail,
    password,
    setPassword,
    handleLogin,
    handleSignup,
    hasAccount,
    setHasAccount,
    emailError,
    passwordError,
  }=props;
```

```
  // const emailRef = useRef();
  // const passwordRef = useRef();
```

```

// const passwordConfirmRef = useRef();
return (
  <Container>
    <Card >
      <Card.Body>
        <h2 className ="text-center ">Login</h2>
      <Form>
        <Form.Group >
          <Form.Label>Email</Form.Label>
          <Form.Control
            type="email"
            required
            autoFocus
            value={email}
            onChange={(e)=> setEmail(e.target.value)}
          />
          <p className="errorMsg">{emailError}</p>
        </Form.Group>
        <Form.Group >
          <Form.Label>Password</Form.Label>
          <Form.Control
            type="password"
            required
            value={password}
            onChange={(e)=> setPassword(e.target.value)}
          />
          <p className="errorMsg" >{passwordError}</p>
        </Form.Group>
      </Container>
      <Button className="w-40" onClick={handleLogin}>Login</Button>
    </Container>
  </Form>
)

```

```

        </Card.Body>
      </Card>
      <div className ="w-100 text-center mt-2">
        Don't have an account? Signup
      </div>
    </Container>
  )
}
export default Login
const Container = styled.div `
  background-color:white;
  margin-bottom:20px;
`

```

Login.css

```

{
  box-sizing: border-box;
  margin: 0;
  padding: 0;
  font-family: "Nunito", sans-serif;
}

button {
  border: none;
  outline: none;
  width: 100%;
  padding: 15px 0;
  color: #fff;
  font-size: 16px;
  letter-spacing: 1px;
  background: #603bbb;
  cursor: pointer;
}

```



```
}

.login {
  width: 100%;
  min-height: 100vh;
  padding: 0 20px;
  background: #e9e9e9;
  display: flex;
}

.login .loginContainer {
  padding: 60px;
  margin: auto;
  width: 100%;
  max-width: 520px;
  min-height: 600px;
  display: flex;
  flex-direction: column;
  justify-content: center;
  background: radial-gradient(
    ellipse at left bottom,
    rgba(22, 24, 47, 1) 0%,
    rgba(38, 20, 72, 0.9) 59%,
    rgba(17, 27, 75, 0.9) 100%
  );
  box-shadow: 0 50px 70px -20px rgba(0, 0, 0, 0.8);
}

.login .loginContainer label {
  color: white;
  margin: 14px 0;
  display: block;
```

```
font-size: 22px;
line-height: 1;
}

.login .loginContainer input {
width: 100%;
border: none;
outline: none;
font-size: 19px;
padding: 10px;
background: rgba(255, 255, 255, 0.1);
color: #fff;
letter-spacing: 1px;
}

.login .loginContainer .btnContainer {
width: 100%;
padding: 24px 0;
}

.login .loginContainer .btnContainer p {
margin: 14px 0 0 0;
text-align: right;
color: #fff;
}

.login .loginContainer .btnContainer p span {
color: yellow;
font-weight: 500;
letter-spacing: 0.5px;
margin-left: 5px;
cursor: pointer;
}
```

```
    transition: all 400ms ease-in-out;
}

.login .loginContainer .btnContainer p span:hover {
    color: red;
}

.errorMsg {
    color: red;
    font-size: 13px;
}

.hero {
    width: 100%;
    min-height: 100vh;
    background: radial-gradient(
        ellipse at left bottom,
        rgba(22, 24, 47, 1) 0%,
        rgba(38, 20, 72, 0.9) 59%,
        rgba(17, 27, 75, 0.9) 100%
    );
}

.hero nav {
    background: #603bbb;
    width: 100%;
    padding: 20px 40px;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
```

```
.hero nav h2 {  
  color: #fff;  
}
```

```
.hero nav button {  
  width: 86px;  
  padding: 8px 0;  
  background: #3a2b58;  
}
```

Signup.js

```
import React, {useRef} from 'react'  
import styled from 'styled-components'  
import { Form,Button,Card} from 'react-bootstrap'  
import './loginstyle.css'
```

```
function Signup(props) {  
  const{  
    email,  
    setEmail,  
    password,  
    setPassword,  
    handleLogin,  
    handleSignup,  
    hasAccount,  
    setHasAccount,  
    emailError,  
    passwordError,  
    name,  
    setName,  
    contact,  
    setContact
```

```
}=props;
```

```
const nameRef = useRef();
const phoneRef = useRef();
const emailRef = useRef();
const passwordRef = useRef();
return (
  <Container>
    <Card>
      <Card.Body>
        <h2 className = "text-center ">Sign Up</h2>
        <Form>
          <Form.Group >
            <Form.Label>Name</Form.Label>
            <Form.Control
              type="text"
              ref={ nameRef }
              value={ name }
              required
              onChange={(e)=> setName(e.target.value)}
            />
            <Form.Label>Phone no.</Form.Label>
            <Form.Control
              type="number"
              ref={ phoneRef }
              value={ contact }
              required
              onChange={(e)=> setContact(e.target.value)}
            />
            <Form.Label>Email</Form.Label>
            <Form.Control
```

```

        type="email"
        ref={emailRef}
        required
        value={email}
        onChange={(e)=> setEmail(e.target.value)}
    />
    <p className="errorMsg">{emailError}</p>
</Form.Group>
<Form.Group >
    <Form.Label>Password</Form.Label>
    <Form.Control
        type="password"
        ref={passwordRef}
        required
        value={password}
        onChange={(e)=> setPassword(e.target.value)}
    />
    <p className="errorMsg">{passwordError}</p>
</Form.Group>
{ /* <Form.Group >
    <Form.Label>Password Confirmation</Form.Label>
    <Form.Control type="password" ref={passwordConfirmRef} required />
</Form.Group> */ }
<Container>
    <Button className="w-40" onClick={handleSignup}>Signup</Button>
</Container>
</Form>
</Card.Body>
</Card>
<div className ="w-100 text-center mt-2">
    Already have an account? Log in
</div>

```

```

    </Container>
  )
}

export default Signup

const Container = styled.div `
  background-color:white;
  margin-bottom:20px;
`

```

Card.js

```

import React from 'react';
import { makeStyles } from '@material-ui/core/styles';
import Accordion from '@material-ui/core/Accordion';
import AccordionSummary from '@material-ui/core/AccordionSummary';
import AccordionDetails from '@material-ui/core/AccordionDetails';
import Typography from '@material-ui/core/Typography';
import ExpandMoreIcon from '@material-ui/icons/ExpandMore';
import './Card.css';

const useStyles = makeStyles((theme) => ({
  root: {
    width: '100%',
  },
  heading: {
    fontSize: theme.typography.pxToRem(15),
    fontWeight: theme.typography.fontWeightRegular,
  },
}));

export default function SimpleAccordion({ flightno,price,
arrival,departure,destination,origin, flight_name, logo,company_website, tt}) {

```

```

const classes = useStyles();
function myFunction() {
  var str = departure;
  var res1 = str.split("T");
  // document.getElementById("demo").innerHTML = res1[0];
  return res1[0]
}

var d1 = new Date(departure)

var d2 = new Date(arrival)
var DepartureTime = `${d1.getUTCHours()}:${d1.getUTCMinutes()===0?'00':
d1.getUTCMinutes()}`
var ArrivalTime = `${d2.getUTCHours()}:${ (d2.getUTCMinutes()===0) ?
`${d2.getUTCMinutes()}0` : d2.getUTCMinutes()}`
var time = `${DepartureTime} - ${ArrivalTime}`
var duration = `${d2.getUTCHours() - d1.getUTCHours()} hr ${ (d2.getUTCMinutes()
- d1.getUTCMinutes())<0?(d1.getUTCMinutes() -
d2.getUTCMinutes()):(d2.getUTCMinutes() - d1.getUTCMinutes())} min`

return (
  <div className={classes.root}>

    <div className="F1">

      <Accordion>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel1a-content"
          id="panel1a-header"
        >
          <div className="flight_container">

```



```

    <a className="logo" title={flight_name} href={company_website}
target="_blank"> <img src={logo}></img></a>
    <div className="time_flight_container">
    <div className="flight_date">{myFunction()}</div>
      <div className="time">{time}</div>
      <div className="flight_name">{flight_name}</div>
    </div>
    <div className="duration_destination_container">
      <div className="duration">{duration}</div>
      <div className="destination">{origin} - {destination}</div>
    </div>
    <div className="stop">Non-Stop</div>
    <div className="select_flight">Select Flight</div>
    <div className="price_trip_container">
      <div className="price">₹ {price}</div>
      <div className="trip">One-Way</div>
    </div>
  </div>

```

```
</AccordionSummary>
```

```
<AccordionDetails>
```

```
<div className="flight_details">
```

```
<div className="line"></div>
```

```
<div className="logo_internal"></div>
```

```
<div className="details">
```

```
<div className="source">
```

```
<div className="dot"></div>
```

```
<div className="source_time">
```

```
<div className="source_arrival_time">{ArrivalTime}</div>
```

```
<div className="dot_1">&nbsp;</div>
```

```
<div className="from">{origin}</div>
```

```
</div>
```

```

    </div>
    <div className="duration_internal">
      <div className="t">Travel Time: </div>
      <div className="travel_time">{duration} hr</div>
    </div>
    <div className="destination_inside">
      <div className="dot"></div>
      <div className="destination_time">
        <div className="destination_arrival_time">{DepartureTime}</div>
        <div className="dot_1"></div>
        <div className="to">{destination}</div>
      </div>
    </div>
    <div className="company">
      <div className="company_code"></div>
      <div className="company_code"></div>
    </div>
    <div className="flight_no"> Flight Number: {flightno}</div>

    </div>

  </div>
</AccordionDetails>
</Accordion>
</div>
</div>
);
}

```

Card.css

```

*{
  margin : 0;

```

```
padding: 0;
}

.F1{
width: 100%;
margin: auto;
margin-top: 20px;
margin-bottom: -10px;
border-radius: 50%;
box-shadow: 0 1px 1px rgba(0,0,0,0.15),
0 2px 2px rgba(0,0,0,0.15),
0 4px 4px rgba(0,0,0,0.15),
0 8px 8px rgba(0,0,0,0.15);
}

.flight_container{
width: 100%;
}

.logo{
width: 3%;
height: 50%;
/* float: right; */
margin-left: -0.5%;
margin-right: 4%;
position: absolute;
top: 40%;
transform: translate(0, -60%);
}

.time_flight_container{
```

```
width: 30%;
float: left;
margin-left: 10%;
}

.duration_destination_container{
width: 30%;
float: left;
}

.stop{
/* width: 8%; */
float: left;
}

.select_flight{
width: 10%;
float: left;
display: none;
}

.price_trip_container{
width: 15%;
float: right;
}

.price{
text-align: right;
font-weight: bolder;
color: green;
}
```

```
.trip{
  text-align: right;
  font-size: 13px;
  color: grey;
}

.flight_date{
  font-weight: bold;
}

.time{
  font-weight: bold;
}

.flight_details{
  width: 100%;
}

.line{
  width: 100%;
  display: block;
  background-color: #dbdbdb;
  height: 1.5px;
}

/* .logo_internal{
  width: 70%;
  display: block;
} */

.details{
```

```
width: 93%;
float: right;
}

.flight_name{
font-size: 13px;
color: rgb(114, 114, 114);
}

.source_arrival_time, .from, .destination_arrival_time, .to, .t, .travel_time, .dot_1{
display: inline-block;
}

.duration_internal{
font-size: 13px;
color: gray;
}

/* .source_time{
display: inline-block;
} */

.travel_time{
padding: 10px;
}

.source_arrival_time, .destination_arrival_time{
margin: auto;
padding-left: 10px;
padding-right: 20px;
}
```

```
.t, .travel_time{
  color: grey;
  padding-left: 10px;
}

.source{
  margin-top: 10px;
}

.destination{
  font-size: 13px;
  color: grey;
}

.flight_no{
  font-size: 13px;
  color: rgb(39, 34, 34);
  margin-top: 10px;
  margin-left: 1%;
}
```

Flights.js

```
import React, {useState, useEffect} from 'react';
import styled from 'styled-components';
import { makeStyles } from '@material-ui/core/styles';
import airline from './airline';
import TextField from '@material-ui/core/TextField';
import Autocomplete from '@material-ui/lab/Autocomplete';
import { KeyboardDatePicker } from '@material-ui/pickers';
import Paper from '@material-ui/core/Paper';
import Fab from '@material-ui/core/Fab';
import SearchIcon from '@material-ui/icons/Search';
import Card from './Card';
```

```
import axios from 'axios';
import {Line} from 'react-chartjs-2';
import fire from '../firebase'
import './Sign/PopupHero'

const useStyles = makeStyles((theme) => ({
  from: {
    width: '300px',
    height:'auto',
    color: 'white',
    margin:'2px',

  },
  container: {
    display:'inline-block',
    width:'inherit',
    padding: '0 5px',
    height:'auto',
  },
  button:{
    margin:'10px',
  },
  date:{
    margin:'2px',
  }
}));

function Flight() {

  var d = new Date(); //date object
```



```

//Location Data
const [origin, setOrigin] = useState("Nagpur");
const [originCityCode, setOriginCityCode] = useState();
const [destinationCityCode, setDestinationCityCode] = useState();
const [destination, setDestination] = useState("Mumbai"); //

//Time data
const [fromTime, setFromTime] = useState(`${d.getDate()}/${(d.getMonth()+1)>
9?(d.getMonth()):("0"+(d.getMonth()+1))}/${d.getFullYear()}`)
const [toTime, setToTime] = useState('27/05/2021');
const [selectedDate, handleDateChange] = useState(new Date());
var currentDate = new Date();
currentDate.setDate(currentDate.getDate() + 1);
const [selectedDate1, handleDateChange1] = useState(currentDate);

//flight Data
const [flightData, setFlightData] = useState([]);
const [guest, setGuest] = useState(1);

const [trip, setTrip] = useState("round");
const TEQUILA_API_KEY = "3mHLBZtsaOzZJB4p58sfIAfxLKMF239G";

const getOriginCityCode= async(city)=>{
  await axios.request({
    method: 'GET',
    url: 'https://tequila-api.kiwi.com/locations/query',
    params: {
      term: origin,
      location_types: 'city',
    },
    headers: {

```

```

    'apikey': 'NZ1N-dUb46M2DP0wrST6VQXyOJ6ndMpm',
  })
  .then(function (response) {
    setOriginCityCode(response.data.locations[0].code)
    console.log(response.data.locations[0].code);
  }).catch(function (error) {
    console.error(error);
  });
}

```

```

const getDestinationCityCode= async(city)=>{
  await axios.request({
    method: 'GET',
    url: 'https://tequila-api.kiwi.com/locations/query',
    params: {
      term: destination,
      location_types: 'city',
    },
    headers: {
      'apikey': 'NZ1N-dUb46M2DP0wrST6VQXyOJ6ndMpm',
    })
  .then(function (response) {
    setDestinationCityCode(response.data.locations[0].code)
    console.log(response.data.locations[0].code);
  }).catch(function (error) {
    console.error(error);
  });
}

```

```

const getDate = (d)=>{
  return `${d.getDate()}/${(d.getMonth()+1)>
9?(d.getMonth()):("0"+(d.getMonth()+1))}/${d.getFullYear()}`

```

```

}

const getData = async () => {
  console.log(origin);
  await getOriginCityCode(origin);
  await getDestinationCityCode(destination);
  setFromTime(getDate(selectedDate));
  setToTime(getDate(selectedDate1));

  await axios.request({
    method: 'GET',
    url: 'https://tequila-api.kiwi.com/v2/search',
    params: {
      fly_from:originCityCode,
      fly_to:destinationCityCode,
      date_from:fromTime,
      date_to: toTime,
      flight_type: "oneway",
      sort:'price',
      max_stopovers:0,
      curr:'INR',
    },
    headers: {
      'apikey': 'NZ1N-dUb46M2DP0wrST6VQXyOJ6ndMpm',
    })
  .then(function (response) {
    setFlightData(response.data.data)
    console.log(response.data);
  }).catch(function (error) {
    console.error(error);
  });
}

```

```
const [chartData, setChartData] = useState({ });

const flightprice = (data) =>{
  let list =[];
  for( const dataObject of data){
    console.log(dataObject.price)
    list.push(dataObject.price)
  }
  return list
}

const flightdate = (data) =>{
  let list1 =[];
  for( const dataObject of data){
    list1.push((dataObject.local_arrival).split('T')[0])

  }
  return list1
}

function ascendingOrder(arr) {
  return arr.sort(function(a, b) {
    return a - b;
  });
}

const getChartData= async ()=>{
  await axios.request({
    method: 'GET',
    url: 'https://tequila-api.kiwi.com/v2/search',
```

```

    params: {
      fly_from:originCityCode,
      fly_to:destinationCityCode,
      date_from:fromTime,
      date_to: toTime,
      flight_type: "oneway",
      sort:'date',
      max_stopovers:0,
      curr:'INR',
    },
    headers: {
      'apikey': 'NZ1N-dUb46M2DP0wrST6VQXyOJ6ndMpm',
    })
  .then(function (response) {
    for(const dataObj of response.data.data){
      flightprice.push(parseInt(dataObj.price))
    };
  }).catch(function (error) {
    console.error(error);
  });
}

const chart =()=>>{
  setChartData({
    labels: flightdate(flightData),
    datasets: [{
      borderColor: "rgba(109,212,0,0.5)",
      pointBorderColor: "#6DD400",
      pointBackgroundColor: "#80b6f4",
      pointHoverBackgroundColor: "#80b6f4",
      pointHoverBorderColor: "#80b6f4",
      pointBorderWidth: 10,

```

```

        pointHoverRadius: 10,
        pointHoverBorderWidth: 1,
        pointRadius: 2,
        fill: false,
        borderWidth: 2,
        data: flightprice(flightData),
    }}
  })
}

```

```

const handleChange = (event) => {
  setTrip(event.target.value);
};

```

```

const setalert=async()=> {
  try{
    let uprice= flightData[0]["price"];
    let ucity1= flightData[0]["cityCodeFrom"];
    let ucity2= flightData[0]["cityCodeTo"];
    let udate1= flightData[0]["local_arrival"];
    let udate2= flightData[0]["local_departure"];
    var uquerycount=0;
    var qcity1="";
    var qcity2="";
    console.log(udate2,udate1)

    const db = fire.firestore();
    console.log(window.useremail);
    await db.collection("users").doc(window.useremail).get().then((snapshot) => {
      let items = snapshot.data();
      var qcount = (items['querycount']);
      console.log(qcount,qcity1,qcity2);
    });
  }
}

```

```

        uquerycount=qcount

    });
    console.log(uquerycount+1);
    uquerycount+=1;

    var query="query"+uquerycount
    if ((ucity1 != qcity1) || (ucity2 != qcity2)){
        await db.collection("users").doc(window.useremail).update({
            [query]:{
                price:uprice,
                city1: ucity1,
                city2: ucity2,
                date1:udate1,
                date2:udate2,},
            querycount:uquerycount,
        });
    }else{
        console.log("repeted");
    }

}catch{

}

// console.log(flightData[0]["price"])
}

const classes = useStyles();
return (
    <FlightPage>
        <Paper className={classes.container} elevation={3}>

```

```

<form action="" onSubmit={ async(event)=>{
  event.preventDefault();
  getData();
  chart();
  setalert();
}}>
<SearchBox>
  <CitySearch>
    <Autocomplete
      className={classes.from}
      freeSolo
      id="From"
      value={origin}
      onChange={(event, newValue) => {
        setOrigin(newValue);
      }}
      options={top100Films.map((option) => option.title)}
      renderInput={(params) => (
        <TextField
          {...params}
          label="From"
          margin="normal"
          variant="outlined"
          InputProps={{ ...params.InputProps, type: 'search' }}
        />
      )}
    />
  <Autocomplete
    className={classes.from}
    freeSolo
    id="From"
    disableClearable

```



```

value={ destination }
  onChange={(event, newValue) => {
    setDestination(newValue);
  }}
options={top100Films.map((option) => option.title)}
renderInput={(params) => (
  <TextField
    {...params}
    label="Destination"
    margin="normal"
    variant="outlined"
    InputProps={{ ...params.InputProps, type: 'search' }}
  />
)}
/>
</CitySearch>
<Datepicker>
<KeyboardDatePicker
  autoOk
  variant="inline"
  inputVariant="outlined"
  label="From Date"
  format="dd/MM/yyyy"
  value={selectedDate}
  onChange={handleDateChange}
  InputAdornmentProps={{ position: "start" }}
  className={classes.date}
/>
<KeyboardDatePicker
  autoOk
  variant="inline"
  inputVariant="outlined"

```

```

        label="To Date"
        format="dd/MM/yyyy"
        value={selectedDate1}
        onChange={handleDateChange1}
        InputAdornmentProps={{ position: "start" }}
        className={classes.date}
    />
</Datepicker>
<Fab className={classes.button} type="submit" color="primary" aria-
label="add">
    <SearchIcon />
</Fab>
</SearchBox>
</form>
</Paper>

<FlightInfo>
    {/* {console.log(flightData)} */}
    {flightData.map((data)=>{
        var comapanay = get_airline(data.route[0].airline);
        return (
            <Card flightno = {data.route[0].flight_no} price={data.price}
departure={tConvert(data.local_departure)} arrival={tConvert(data.local_arrival)}
origin={origin} destination={destination} flight_name={comapanay[0]}
logo={comapanay[1]} company_website={comapanay[2]}/>
        )
    })}
    {/* {console.log(flightData)} */}
</FlightInfo>
<Paper style={{ paddingLeft:"10px", paddingRight:"10px",marginTop:"20px"
}}>

```

```

    <div style={{ width:"700px"}}>
    <Line data={chartData} />
    </div>

    </Paper>
  </FlightPage>
)
}

```

```
export default Flight
```

```

const FlightPage = styled.div `
display:flex;
flex-direction:column;
align-items:center;
padding:10px;
width: 100%;
`

```

```

const FlightInfo = styled.div `
margin-top:10px;
width:85%;
height:450px;
overflow-y:scroll;
::-webkit-scrollbar{
  display:none;
}
`

```

```

const SearchBox = styled.div `
display:flex;
justify-content:center;
align-items:center;
`

```

```

const DatePicker = styled.div `
margin-top:8px;
display:flex;
`

const CitySearch = styled.div `
display:flex;
margin:10px;
`;

const top100Films = [
  { title: 'Nagpur', year: 1994 },
  { title: 'New-delhi', year: 1972 },
  { title: 'Raipur', year: 1974 },
  { title: 'Mumbai', year: 1974 },
  { title: 'Bangalore', year: 1974 },
  { title: 'Chennai', year: 1974 },
  { title: 'London', year: 1974 },
  { title: 'Ladakh', year: 1974 },

];

function get_airline(airline_code){
  var comapanay_name = "";
  var company_logo = "";
  var company_website="";
  switch(airline_code){
    case 'AI':
      comapanay_name = "Air India";
      company_logo = "https://www.gstatic.com/flights/airline_logos/70px/AI.png";
      company_website="http://www.airindia.in/";
  }
}

```

```
break;
case 'UK':
    comapany_name = "Vistara";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/UK.png";
    company_website="https://www.airvistara.com/in/en";
    break;
case 'IX':
    comapany_name = "Express India";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/IX.png";
    company_website="https://www.airindiaexpress.in/en";
    break;
case 'SG':
    comapany_name = "Spice Jet";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/SG.png";
    company_website="https://www.spicejet.com/";
    break;
case 'G8':
    comapany_name = "Go Air";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/G8.png";
    company_website="https://www.flygofirst.com/";
    break;
case '6E':
    comapany_name = "IndiGo";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/6E.png";
    company_website="https://www.goindigo.in/";

    break;
case 'I5':
    comapany_name = "AirAsia India";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/I5.png";
    company_website="https://www.airasia.co.in/home";
    break;
```

```
case '9I':
    comapany_name = "Alliance Air";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/9I.png";
    company_website="http://www.airindia.in/alliance-air.htm";
    break;
case '4H':
    comapany_name = "Air Hertiage";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/4H.png";
    company_website="https://www.alternativeairlines.com/air-heritage";
    break;
case 'S9':
    comapany_name = "FlyBig";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/S9.png";
    company_website="https://flybig.in/book";
    break;
case 'OG':
    comapany_name = "Star Air";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/OG.png";
    company_website="https://starair.in/";
    break;
case '2T':
    comapany_name = "TruJet";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/2T.png";
    company_website="https://www.trujet.com/#/home";
    break;
case 'ZO':
    comapany_name = "Zoom Air";
    company_logo = "https://www.gstatic.com/flights/airline_logos/70px/ZO.png";
    company_website="https://zoomair.in/";
    break;
}
return [comapany_name, company_logo, company_website];
```

```

}

function tConvert (time) {
  // Check correct time format and split into components
  time = time.toString ().match (/^([01]\d|2[0-3])(:)([0-5]\d)(:[0-5]\d)?$/ ) || [time];

  if (time.length > 1) { // If time format correct
    time = time.slice (1); // Remove full string match value
    time[5] = +time[0] < 12 ? 'AM' : 'PM'; // Set AM/PM
    time[0] = +time[0] % 12 || 12; // Adjust hours
  }
  return time.join (""); // return adjusted time or original string
}

```

Flights.css

```

*{
  margin : 0;
  padding: 0;
}

.heading {
  text-align: center;
  font-weight:bold;
  margin-top: 0;
  font-size: 50px;
  /* text-shadow: 0px 2px 4px rgb(192, 73, 73); */
  font-family: 'Noto Sans JP', sans-serif;
}

```

Hero.js

```

import React from 'react';
import { makeStyles, withStyles } from '@material-ui/core/styles';

```

```

import Tabs from '@material-ui/core/Tabs';
import Tab from '@material-ui/core/Tab';
import Typography from '@material-ui/core/Typography';
import styled from 'styled-components';
import Flight from './Tabs/Flight';
import Experience from './Tabs/Experience';
import Hotels from './Tabs/Hotels';

const StyledTabs = withStyles({
  indicator: {
    display: 'flex',
    justifyContent: 'center',
    backgroundColor: 'transparent',
    '& > span': {
      maxWidth: 40,
      width: '100%',
      backgroundColor: 'white',
    },
  },
})(
  (props) => <Tabs {...props} TabIndicatorProps={{ children: <span /> }} />);

const StyledTab = withStyles((theme) => ({
  root: {
    textTransform: 'none',
    color: 'white',
    fontWeight: theme.typography.fontWeightRegular,
    fontSize: theme.typography.pxToRem(15),
    marginRight: theme.spacing(1),
    '&:focus': {
      opacity: 1,
    }
  }
}));

```



```

    },
  },
  ))((props) => <Tab disableRipple {...props} />);

const useStyles = makeStyles((theme) => ({
  root: {
    flexGrow: 1,
  },
  padding: {
    padding: theme.spacing(0),
  },
  demo1: {
    backgroundColor: theme.palette.background.paper,
  },
  demo2: {
    backgroundColor: '#2e1534',
  },
  demo3: {
    backgroundColor: 'black',
  }
}));

export default function CustomizedTabs() {
  const classes = useStyles();
  const [value, setValue] = React.useState(0);

  const handleChange = (event, newValue) => {
    setValue(newValue);
  };

  return (
    <div className={classes.root}>

```

```

    <div className={classes.demo3}>
      <StyledTabs value={value} onChange={handleChange} aria-label="styled tabs
example" centered>
        <StyledTab label="Flights" />
        <StyledTab label="Experiences" />
        <StyledTab label="Hotels" />
      </StyledTabs>
      <TabPanel>
        <BackgroundImage>
          <Span>
            { value === 0 && <Flight/>}
            { value === 1 && <Experience/>}
            { value === 2 && <Hotels/>}
          </Span>
        </BackgroundImage>
      </TabPanel>
    </div>
  </div>
);
}

```

```

const TabPanel = styled.div `
width:100%;
height:calc(100vh + 100px);
background-color:black;
margin-top:70px;
`

```

```

const Span = styled.div `
margin-top:-60px;
`

```

```
const BackgroundImage = styled.div `
background-image:url('/airplane.jpg');
background-size:cover;
display:flex;
justify-content:center;
width:95%;
margin-top:20px;
height:600px;
margin-left:auto;
margin-right:auto;
`
```

Main.py

```
from datetime import datetime, timedelta
import requests
```

```
From=input("FROM: ").title()
Destination=input("DESTINATION: ").title()
months=int(input("how many months from now you want to search for: "))
trip=input("ways: ").lower()
```

```
TEQUILA_ENDPOINT = "https://tequila-api.kiwi.com"
TEQUILA_API_KEY = "NZ1N-dUb46M2DP0wrST6VQXyOJ6ndMpm"
```

```
def get_destination_code(self, city_name):
    location_endpoint = f"{TEQUILA_ENDPOINT}/locations/query"
    headers = {"apikey": TEQUILA_API_KEY}
    query = {"term": city_name, "location_types": "city"}
    response = requests.get(url=location_endpoint, headers=headers, params=query)
    results = response.json()["locations"]
```

```

code = results[0]["code"]
return code

origin_city_code= get_destination_code(0,From)
destination_city_code= get_destination_code(0,Destination)
print (origin_city_code)
print (destination_city_code)

from_time = datetime.now() + timedelta(days=1)
to_time = datetime.now() + timedelta(days=(months * 30))

headers = {"apikey": TEQUILA_API_KEY}
query = {
    "fly_from": "BOM",
    "fly_to": "NAG",
    "date_from": from_time.strftime("%d/%m/%Y"),
    "date_to": to_time.strftime("%d/%m/%Y"),
    # "nights_in_dst_from": 7,
    # "nights_in_dst_to": 28,
    "flight_type": "oneway",
    "one_for_city": 1,
    "max_stopovers": 0,
    "curr": "INR"
}
response = requests.get(
    url=f"{TEQUILA_ENDPOINT}/v2/search",
    headers=headers,
    params=query,
)
data = response.json()["data"][0]
print(data)
#

```

```
price=data["price"]
local_arrival=data["local_arrival"]
out_date=data["route"][0]["local_departure"].split("T")[0],
out_time=data["route"][0]["local_departure"].split("T")[1],

# out_date=data["route"][0]["local_departure"].split("T")[0],
# return_date=data["route"][1]["local_departure"].split("T")[0]
# print(out_date)
# print(return_date)
print(price)

# departure date and time flight
print(out_date)
print(out_time)

# time and date when you reach to the destination
print(local_arrival)
```

TESTING AND VALIDATION

Testing & Validation

System Testing

The personnel in the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. The success or failure of well- designed and technically elegant systems can depend on the way they are operated and used.

Technique Used in Testing:

1. Unit Testing:

In this testing, we need to focus on the individual program and does not need to test the whole program. Unit testing focuses on the module independently. After developing each module we test it individually to determine whether the module is satisfactory or not.

2. Program Testing:

Under program testing, we have to tested the actual output with the expected output. Program testing check the two types of errors: syntax and logic. A syntax error is program statement that violates the one or more rules of the programming languages. Logic error deals with the incorrect data fields, out of range items and invalid combination.

3. Integration Testing:

Integration testing aims at constructing the program structure while at the same constructing tests to uncover error associated with interfacing the modules are integrated by using the top down approach.

4. Document Testing:

Document Testing is necessary for the project .It tries to find out whenever document supplies are satisfactory or any further document should be supplied. In this project, we done the documentation testing . So, the entire document which is applied with project is satisfactory.

Validation

Validation is nothing but the security measures taken at the time of execution of any program. It is necessary for the analyst to take validation in their project as it provides more accuracy and systematic flow to the project. Validation not only stops inputs of the false data but also the role of guide during input of data.

Validation put its control over the data. In both characters as well as integer data type, whenever wrong data or invalid data is stored by the user it fires a message immediately and warns the user to input correct data type. Validation testing ensures that the product actually meet the client's requirements. It can also be defined to demonstrate that the product fulfils its intended use when deployed on appropriate environment.

The validations provided on our software are:

1. Error messages:

In our system we have provided all validations that will guide the user how perform operations when the user will enter incorrect data into system, user will not get desired output. To avoid this problem, we have introduced "errors messages" to guide the user to enter correct data into the system.

2. Duplication of values:

In our project, we provide the validation for duplicate values. When the user entered the valued which is similar to the existing value in the database then the message will be displayed that there is duplication of data.

3. Password:

In our project, during the time of registration and login the user must entered the password in character type and length is eight only. So, the user must create the password not only in character data type but also length size is eight.

**SYSTEM SECURITIES AND
MEASURES**

System Security measures are-

Firewall

One widely used strategy to improve system security is to use a firewall. A firewall consists of software and hardware set up between an internal computer network

and the Internet. A computer network manager sets up the rules for the firewall to filter out unwanted intrusions. These rules are set up in such a way that unauthorized access is much more difficult.

Passwords

The most widely used method to prevent unauthorized access is to use passwords. A password is a string of characters used to authenticate a user to access a system.

The password needs to be kept secret and is only intended for the specific user. In computer systems, each password is associated with a specific username since many individuals may be accessing the same system.

Good passwords are essential to keeping computer systems secure.

Unfortunately, many computer users don't use very secure passwords, such as the name of a family member or important dates - things that would be relatively easy to guess by a hacker. One of the most widely used passwords - you guessed it 'password' Definitely not a good password to use.

**IMPLEMENTATION, EVALUATION
AND MAINTENANCE**

System Implementation

After you have carefully planned your project , you will be ready to start the project implementation phase.

The implementation phase involves putting the project plans into action. Implementation as an activity has to be carefully managed.

The System implementation stands for the conversion of three types-

- 1.System conversion of manual system into computerized system in the way to understand by the user of the project made by us being access.
- 2.Conversion of existing computerized system into modified version of hardware and software both are checked by us for better performance.
- 3.Keeping the hardware and implementing the new techniques is where we checked other hardware i.e., Ram, Hard disk for better performance.

System Evaluation

Dynamic pricing is a technique of pricing a product according to current market conditions. Prices change in real time based on timely data: Data about customer booking patterns, competitor prices, even weather and popular events can impact the product demand and require you to adjust prices to increase profits. This practice is most valuable to products that share these characteristics:

Evolutions

After the MIS has been operating smoothly for a short period of time, an evaluation of each step in the design and of the final system performance should be made. Evaluation should not be delayed beyond the time when the system's analysts have completed most of the debugging. The longer the delay, the more difficult it will be for designer to remember important details.

The evaluation should be made by the customer as well as by the designers.

Maintenance

Control and maintenance of the system are the responsibilities of the line managers. Control of the systems means the operation of the system as it was designed to operate. Sometimes, well-intentioned people or operators may make unauthorized changes to improve the system, changes that are not approved or documented.

Maintenance is closely related to control. Maintenance is that ongoing activity that keeps the MIS at the highest levels of effectiveness and efficiency within cost constraints.

Maintenance is directed towards reducing errors due to design, reducing errors due to environmental changes and improving the system's scope and services.

FUTURE SCOPE

Future Scope of Project

A flight price tracker is an effective way to track our flight and book it when the price changes. It is a must-have tool for every traveler who is willing to have a ticket on budget and a smooth search for a flight from one platform. Thus Adding more functionality for users will make it more convenient for them.

Functionality such as It is also possible to have multiple price alerts set up at once which is a great option if you are comparing vacation destinations, sorting the price for round trips, to book the airway ticket adding payment integration which will be more helpful for the customer.

With this functionality, it will also contain the search for a hotel at your desired location. Your final goal is to build a Complete tour guide for budget travelers.

CONCLUSION

CONCLUSION

The flight price tracker recommends the best time to buy an airway ticket. When the price changes, customers are notified via email or SMS if the price went up or down (and by how much). Flight alerts are completely free and can be stopped at any moment. It is also possible to have multiple price alerts set up at once which is a great option if you are comparing vacation destinations. It really is a must-have tool, especially for budget travelers, because flight alerts are hands-down one of the best ways to find cheap flights quickly.

Reference and Bibliography

Reference:

Book Reference:

1)PHP Black Book

2)System Analysis and Design

-Elias Awad

(Galgotia publications Pvt.)

[1] O. Etzioni, R. Tuchinda, C. A. Knoblock, and A. Yates. To buy or not to buy: mining airfare data to minimize ticket purchase price.

[2] Manolis Papadakis. Predicting Airfare Prices.

[3] Groves and Gini, 2011. A Regression Model For Predicting Optimal Purchase Timing For Airline Tickets.

[4] Modeling of United States Airline Fares – Using the Official Airline Guide (OAG) and Airline Origin and Destination Survey (DB1B), Krishna Rama-Murthy, 2006.

[5] B. S. Everitt: The Cambridge Dictionary of Statistics, Cambridge University Press, Cambridge (3rd edition, 2006). ISBN 0-521-69027-7.

[6] E. Bachis and C. A. Piga. Low-cost airlines and online price dispersion. International Journal of Industrial Organization, In Press, Corrected Proof, 2011.

Bibliography –

While developing this project internet was the eternal support. Following are the websites referred by us which helped us in developing our project :

<https://www.w3schools.com/php/>

<https://www.geeksforgeeks.org/>

<https://www.javatpoint.com/>

Approved Copy Of Synopsis

A
PROJECT SYNOPSIS
ON

“Cheapest Flight Price Tracker”

Submitted to

**G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
AUTONOMOUS**

In the Partial Fulfillment of

B.Com. (Computer Application) Final Year

Synopsis Submitted by

Atharva Pande

Atharva Kausadikar

Under the Guidance of

Pravin J. Yadao



**G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
AUTONOMOUS
2021-2022**

1. Introduction: (Write 4 to 5 lines)

“Cheapest Flight price tracker” recommend the best time to buy airway ticket. When the price changes, you'll be notified. Flight alerts are completely free. It shows different price as per user's budget or with different flight option according to vacation destinations or business tours. It is one of the best ways to find cheap flights quickly.

2. Objectives of the project: (Write only 5 points)

- 1) To track flight.
- 2) To send the notifications about the cheapest flight and also alert the user when there is sudden price drop.
- 3) To provide flight ticket at Economical or Affordable price.
- 4) To get an intuition of future flight prices.
- 5) To reduce customer/users time and save money by providing services without any brokerage.

3. Project Category: Webpage

4. Tools/ Platform/ Languages to be used: MySQL, Python, Html, CSS, JavaScript,

5. Scope of future application: (Write 4 to 5 points)

- 1) We can add hotels booking facility
- 2) Prediction of future flight price.
- 3) Short Information related to destination or Famous touring places.
- 4) Proper Travel planning information according to reliable timing for users.
- 5) Customer Reviews for better improvement with FAQs and their answers

Submitted by,

Atharva Pande

Atharva Kausadikar

Approved by,

Prof. Pravin Yadao

Project Guide