

## [AIRLINES RESERVATION SYSTEM]

*A Minor project submitted for the partial fulfilment of the requirements for the degree of*  
**MASTER OF COMPUTER APPLICATION**

*Submitted By*

1. Ishita Naskar Roll No: 18871021054 Reg. No.: 211880571010019 of 2021-2023
2. Prarthana Ghosh Roll No: 18871021049 Reg. No.: 211880571010020 of 2021-2023
3. Subhajit Naskar Roll No: 18871021057 Reg. No.: 211880571010057 of 2021-2023

*Under The Supervision Of*

**Mr. Subhojit Santra**

Asst. Professor

**Mr. Tapan Kumar Chakrabarty**

HOD MCA

Department Of MCA

Techno India (Hooghly – Campus)



Techno India (Hooghly – Campus)  
Dharampur, G.T. Road, Near Khadina More  
Chinsurah, Hooghly-712101

Website: [www.technoindiahooghly.org](http://www.technoindiahooghly.org), Email: [info@technoindiahooghly.org](mailto:info@technoindiahooghly.org)

**Project Undertaken:**

# FLY HIGH





## Techno India Hooghly - Campus

Dharampur, Shantiniketan, Near Khadinamore

G. T. Road, Chinsurah, Hooghly - 712101

Phone: (033) 2680-2389 / 6565

E-mail: [info@technoindiahoghly.org](mailto:info@technoindiahoghly.org)

Website: [www.technoindiahoghly.org](http://www.technoindiahoghly.org)

### Certificate

*This is to certify that the project entitled “**AIRLINES RESERVATION SYSTEM**” submitted by **Ishita Naskar**, University Roll No: **18871021054** and Registration No.: **211880571010019**, **Prarthana Ghosh**, University Roll No: **18871021049** and Registration No.: **211880571010020**, **Subhajit Naskar**, University Roll No: **18871021057** and Registration No.: **211880571010057** of 2021-23 as Assigned Project (minor) for the partial fulfilments of degree of **Master of Computer Application** is worth of acceptance.*

*Subhojit Santra*

**Mr. Subhojit Santra**

(Assistant Professor)

**Mr. Tapan kr Chakrabarty**

(Head of The Department, MCA)

Techno India Hooghly

### ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed to the development of this work and who influenced our thinking, behaviour, and acts during the course of study.

We express sincere gratitude to **Dr. Mrinal Kanti Chakrabarty** worthy **Principal** for providing us an opportunity to undergo this project at **Techno India (Hooghly-Campus)**.

. We are grateful to our project guide “**Mr. Subhojit Santra & Mr. Tapan Kr. Chakraborty**” for the guidance, inspiration and constructive suggestions that helpful us in the preparation of this project.

We are also thankful to my group with whom we have fruitful discussions which have helped us a lot in giving a final shape to the program.

#### PROJECT ASSOCIATES:

**Prarthana Ghosh (18871021049)**

**Ishita Naskar (18871021054)**

**Subhajit Naskar (18871021057)**

# ABSTRACT

The purpose of the project entitled as “ **AIRLINES RESERVATION SYSTEM** ” is to computerize the Front Office Management of Airport to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of passengers information, journey details, etc. Traditionally, it was done manually. The main function of the system is register and store passengers’ details and their others detail and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains passengers’ details, journey details, while system output is to get these details on to the screen. The **AIRLINES RESERVATION SYSTEM** can be entered using a username and password. It is accessible either by an administrator or passengers. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

# PROJECT REPORT

## INDEX

Sr. No.	Table OF Content	Page No
1.)	Project name and description	4
2.)	Profile OF The Problem	5
3.)	Existing System	6-7
	a.) Introduction	
	b.) Existing Software	
	c.) What's new in the system to be developed?	
4.)	Problem Analysis	8-10
	a.) Product Definition	
	b.) Feasibility Analysis	
	c.) Project Plan	
5.)	Software Requirement Analysis	10-11
	a.) Introduction	
	b.) Specific Requirement	
6.)	Design	11-20
	a.) System Design	
	b.) Detailed Design	
7.)	Testing	20
	a.) Functional Testing	
	b.) Structural Testing	
	c.) Levels of testing	
	d.) Testing of the Project	
8.)	implementation	24
	a.) Implementation of the project	
9.)	User Manual	27
	a.) The complete guide for the project developed	
10.)	Source code	27
11.)	bibliography	40

# PROJECT REPORT

## 1. Project Name and Description

### AIRLINES RESERVATION SYSTEM



The Airline Reservations System (ARS) was one of the earliest changes to improve efficiency. ARS eventually evolved into the Computer Reservations System (CRS). A Computer Reservation System is used for the reservations of a particular airline and interfaces with a Global Distribution System (GDS) which supports travel agencies and other distribution channels in making reservations for most major airlines in a single system.

Airline Reservations Systems contain airline schedules, fare tariffs, passenger reservations and ticket records. An airline's direct distribution works within their own reservation system, as well as pushing out information to the GDS. A second type of direct distribution channel are consumers who use the internet or mobile applications to make their own reservations.

### INVENTORY MANAGEMENT

An airline's inventory contains all flights with their available seats. The inventory of an airline is generally divided into service classes (e.g., First, Business or Economy class) and up to 26 booking classes, for which different prices and booking conditions apply. Inventory data is imported and maintained through a Schedule Distribution System over standardized interfaces. One of the core functions of the inventory management is the inventory control. Inventory control steers how many seats are available in the different booking classes, by opening and closing individual booking classes for sale. In combination with the fares and booking conditions stored in the Fare Quote System the price for each sold seat is determined.

## DISPLAY AND RESERVATION

Users access an airline's inventory through an availability display. It contains all offered flights for a particular city-pair with their available seats in the different booking classes. This display contains flights which are operated by the airline itself as well as code share flights which are operated in co-operation with another airline. If the city pair is not one on which the airline offers service, it may display a connection using its' own flights or display the flights of other airlines.

The availability of seats of other airlines is updated through standard industry interfaces. Depending on the type of co-operation it supports access to the last seat (Last Seat Availability) in real-time. Reservations for individual passengers or groups are stored in a so-called Passenger Name Record (PNR). Among other data, the PNR contains personal information such as name, contact information or special services requests (SSRs).

e.g., for a vegetarian meal, as well as the flights (segments) and issued tickets. Some reservation systems also allow to store customer data in profiles to avoid data re-entry each time a new reservation is made for a known passenger.

### Fare Quote and Ticketing

The Fares data store contains fare tariffs, rule sets, routing maps, class of service tables, and some tax information that construct the price - "the fare". Rules like booking conditions (e.g. minimum stay, advance purchase, etc.) are tailored differently between different city pairs or zones, and assigned a class of service corresponding to its appropriate inventory bucket. Inventory control can also be manipulated manually through the availability feeds, dynamically controlling how many seats are offered for a particular price by opening and closing particular classes.

## 2. Profile of the problem

The web based "airline reservation system" project is an attempt to stimulate the basic concepts of airline reservation system. The system enables the customer to do the things such as search for airline flights for two travel cities on a specified date, choose a flight based on the details, reservation of flight and cancellation of reservation.

The system allows the airline passenger to search for flights that are available between the two travel cities, namely the "Departure city" and "Arrival city" for a particular departure and

# PROJECT REPORT

arrival dates. The system displays all the flight's details such as flight no, name, price and duration of journey etc.

After search the system display list of available flights and allows customer to choose a particular flight. Then the system checks for the availability of seats on the flight. If the seats are available then the system allows the passenger to book a seat. Otherwise, it asks the user to choose another flight.

To book a flight the system asks the customer to enter his details such as name, address, city, state, credit card number and contact number. Then it checks the validity of card and book the flight and update the airline database and user database.

## **Disadvantages in existing system:->**

- Time consuming
- Possibly of losing data
- Lack of security
- Difficulties in maintaining records
- Human error will be frequent
- searching the records manually leads time consuming

## **3. Existing System**

### **Introduction**

In the existing system all the data are stored manually to an excel sheet and filed accordingly in a filing cabinet.

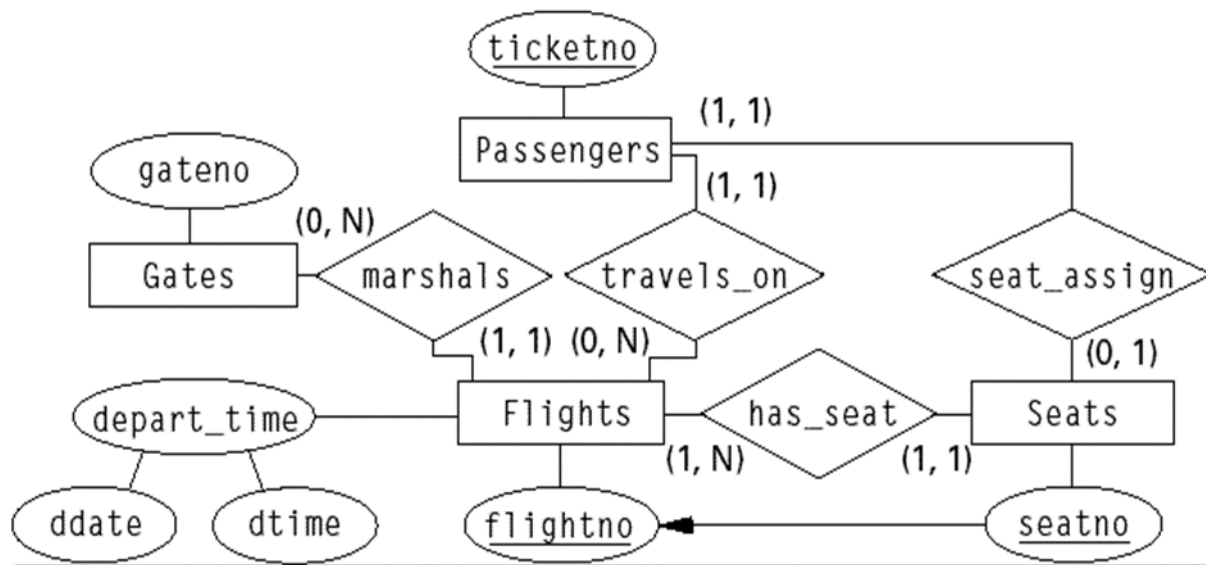
### **SPECIFIC OBJECTIVES: -**

- a. Maintaining safe records
- b. Should be easier to find for a record in the database
- c. The loss of data in any corruption of any files in the system will be avoided due to any natural case
- d. Minimizing errors of the information recorded in the system while entering to the system
- e. The new system should be user friendly
- f. Data entry should be fast
- g. There should be a method of keeping the information from unauthorized users.



# PROJECT REPORT

## ERD



### Methods of data collection and input Structure of Airline:

Name:

Address:

Phn Number:

Dob:

Nic:

Passport No:

Nationality:

Destination:

Airline:

Day of Dep:

Time:


When the customer asks to book a flight at first the booking form is been filled with his/her full:->

# PROJECT REPORT

Name, Address, National ID, Email, Address, Contact Number, Destination, booking date and the Return date as shown above the form. After that the document is signed by the office staff.

✚ Exiting software

## METHODOLOGIES

### HARDWARE:

PROCESSOR: PENTIUM IV 2.6 GHz

RAM: 512MB DD RAM

MONITOR: 15" COLOR

HARD DISK :250 GB

CDDRIVE: LG52X

KEYBOARD: STANDARD 102 KEYS

MOUSE: OPTICAL MOUSE

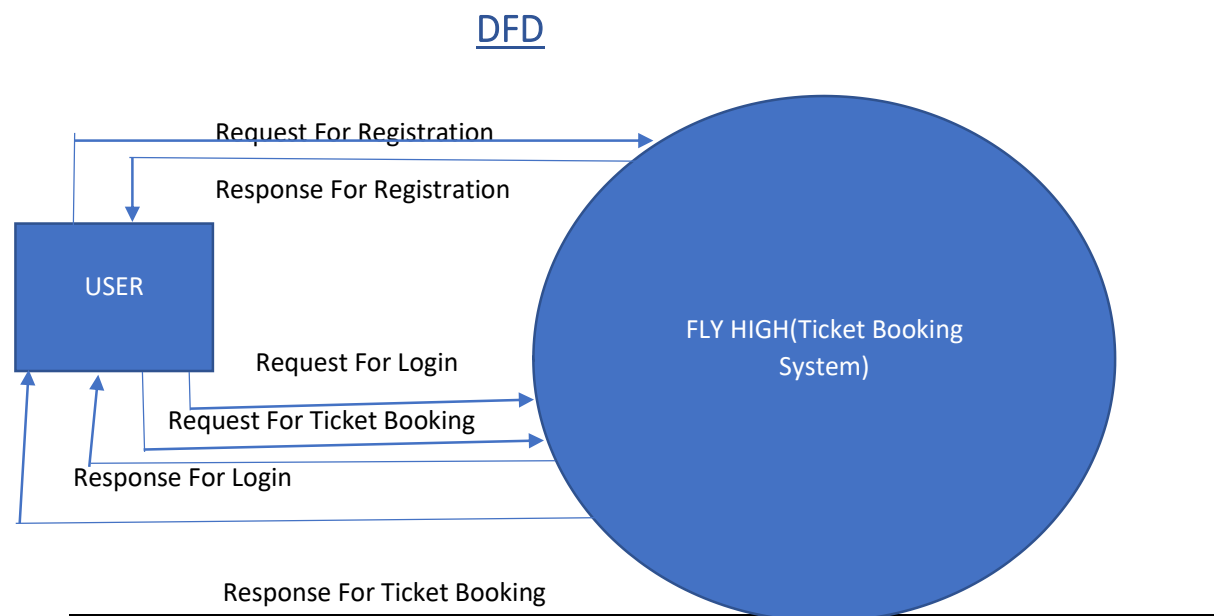
### SOFTWARE:

FRONT END: HTML, CSS, JAVA SCRIPT, REACT

BACKEND:

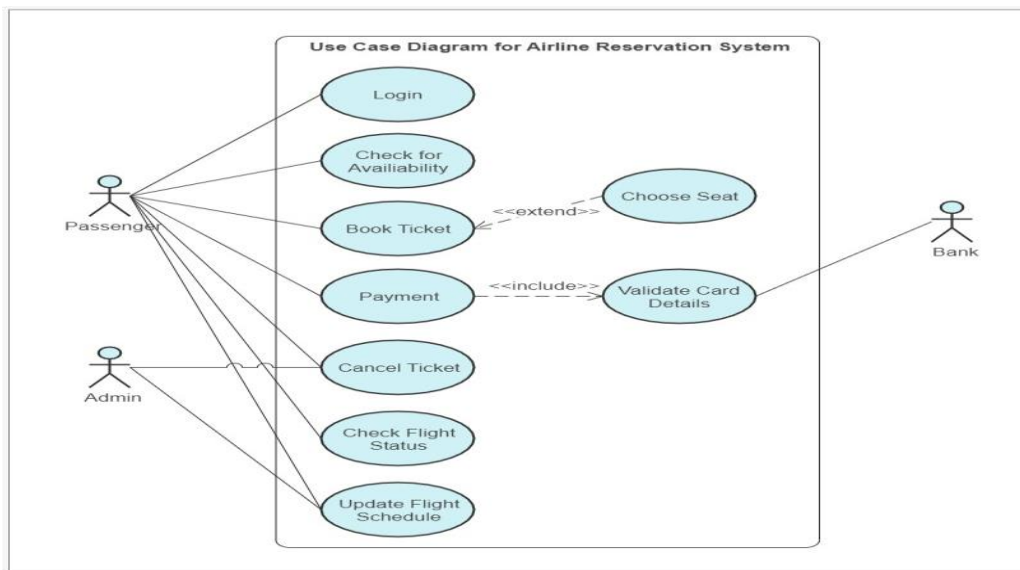
OPERATING SYSTEM: WINDOWS, WINDOWS XP

✚ DFD for Present system



# PROJECT REPORT

## • USECASE DIAGRAM: -



## GNATT CHART: -

Task	Jan '20	Feb '20	Mar '20	Apr '20	May '20	Jun '20	Jul '20	Aug '20
Planning								
Research								
Design and development								
Implementation								
Follow-up and action								

## What's new in the system to be developed?

### APPLICABILITY

- This project solves the problem of the traditional reservation system.
- With certain changes it can be applicable on any online reservation field.
- One of the most benefits in today's life is that reservation can be made from any place of the world.
- The user need not to be present the physically to draw a reservation slip. It will automatically do by the system

### ADVANTAGES

# PROJECT REPORT

1. It easy to learn and adjust to the system
2. this system does not require the staff to be highly educated
3. the requirements to tackle this job may be limited to
4. Willing to work long hours
5. data is not easily lost
6. it easy to manage the system due to the high number of staff working

## 4. Problem Analysis

### **Product definition**

1. Plane **type**: This defines the physical type of the plane. It dictates the capacity of first, executive, business and economy seats that a flight can have.
2. **Airport**: An airport consists of a name, the city it is in, and its airport id.
3. **Flight**: A flight is identified by its flight ID. A flight denotes a unique “plane”, i.e., one which is scheduled to run at a certain time, from one place to another. A flight runs over a set of routes.
4. **Route**: A route is simply a tuple of airports: (Start Airport, End Airport), and every route has a unique route id. A flight runs over a route only if it runs from the start airport to the end airport, possibly halting in between at other airports. A route is elementary for a flight if the flight runs nonstop from the start airport to the end airport.
5. **Ticket**: A ticket is uniquely identified by a ticket ID. The ticket may be a passenger ticket or a cargo ticket, and can be booked under a passenger profile or a user profile. A ticket is booked on a flight for a route that the flight is associated with. A passenger ticket contains details about the passenger, and a cargo ticket about a cargo. The
6. **Scheme**: A scheme consists of discount percentages on various classes awarded on certain flights, and for certain people or round trips. Scheme ids have a type code defining what they they are valid for, and a period code showing whether they are valid as of now or no. A scheme is defined for a flight and for a particular route.
7. **Official**: An official is a person who can book tickets for others, and can find retrieve the complete list of passengers boarding a flight. An official works at an airport.
8. **Profile**: A profile denotes that a person has been verified to be genuine and can book tickets/ execute certain queries.

# PROJECT REPORT

## Feasibility Analysis

1. **Flight Route Scheme**: This is a ternary relation that says that a flight runs over a route using a particular scheme. The scheme can be null, but not the flight id and route id. The attributes in this include:

- a. Fare for the flight between the two stops given by the route specified.
- b. Any scheme valid on this flight, for this route
- c. Number of booked seats on this flight, route.
- d. A flag value indicating whether this is an elementary or complex route for this flight.

2. **Flight Route Ticket**: This ternary relation says that a ticket is booked on a certain flight over one route that the flight allows. It has only the primary keys of each entity.

3. **Profile Ticket relation**: This consists of two relations: user profile related to ticket and the official profile related to ticket. This is done to keep the user and official profiles separate and independent from each other.

## Project Plan

**The plan of this project is to make a software which is:->>**

- Interpreted and high performance
- Distributed
- Dynamic
- Secure

### **Services Reservation in Airlines Abstract: -**

This Abstract report on a study about examining airlines' reservation services.

An apparatus, method, and program for determining a price of an option to purchase an airline ticket, and for facilitating the sale & exercise of those options.

Pricing of the options may be based on departure location criteria, destination location criteria, and travel criteria. The attributes selected for examination included....

- (1) requirements of reservation services.
- (2) provision of extra benefits.
- (3) factors affecting reservation time.
- (4) provision of additional services/facilities.

Empirical results indicated that some airlines did not provide all components in the chosen attributes and that airlines in these regions differed significantly in certain dimensions of the chosen attributes.

## 5. Software requirement Analysis

### Introduction

Requirements analysis is critical to the success of a development project. Requirements must be documented, actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Requirements can be architectural, structural, behavioural, functional, and non-functional.

### General Description

**VS CODE** - Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE

### Specific Requirements

#### ▪ System

- Windows XP Service Pack 2 or above (for 2010 release, Service Pack 3)
- Windows Server 2003 Service Pack 1 or above
- Windows Server 2003 R2 or above
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows 10
- Linux with Mono (only works with .NET 2.0 applications)

#### ▪ Hardware

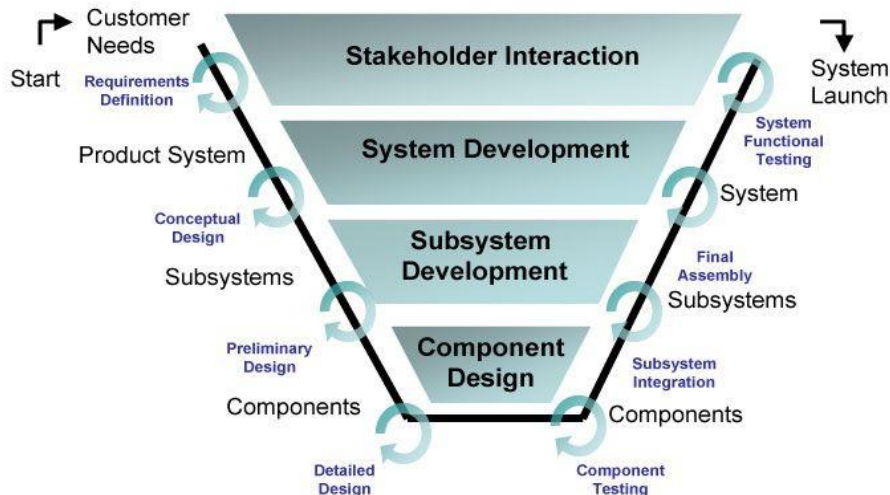
- Minimum: 1.6 GHz CPU, 384 MB RAM, 1024×768 display, 5400 RPM hard disk
- Recommended: 2.2 GHz or higher CPU, 1024 MB or more RAM, 1280×1024 display, 7200 RPM or higher hard disk

# PROJECT REPORT

## 6. Design

### ✚ System Design

**Systems design** is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.



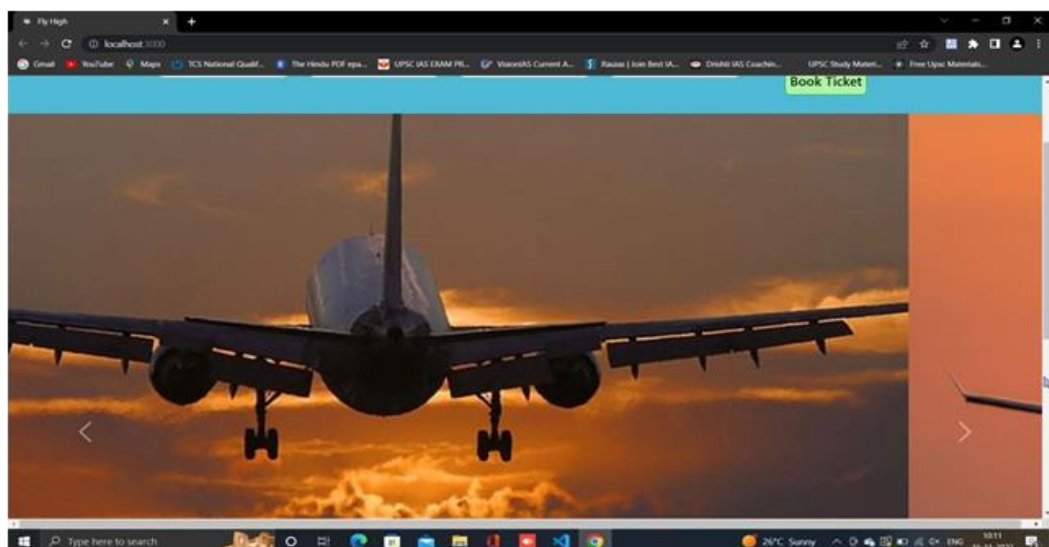
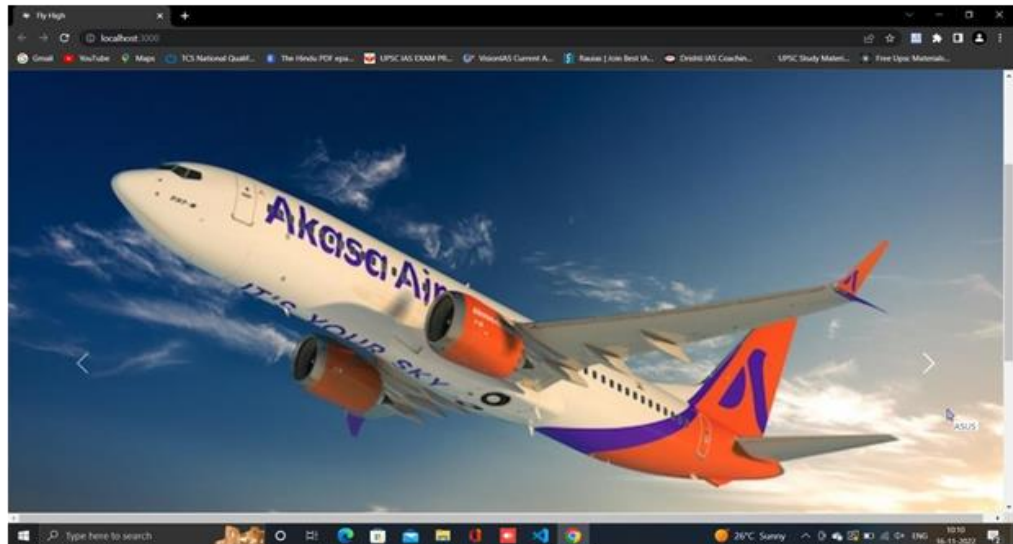
There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development, then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user

# PROJECT REPORT

## Detailed design

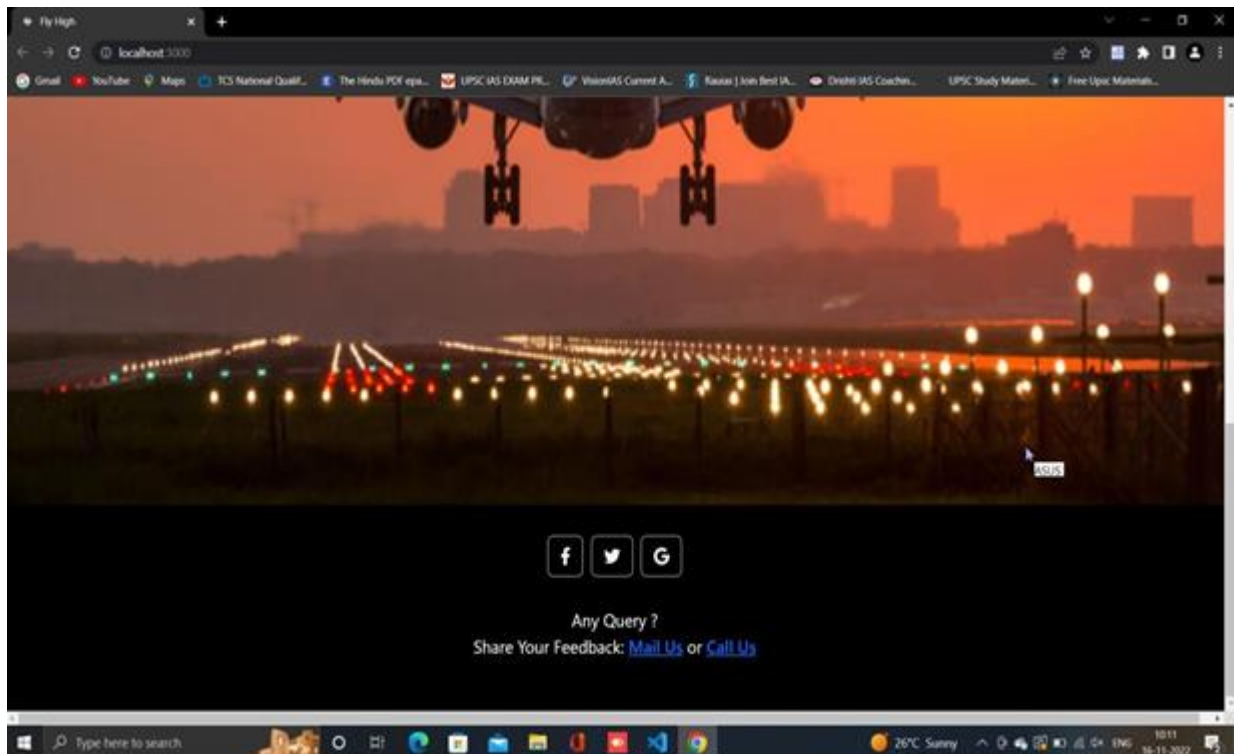
Screen shots:

### Design Screenshot

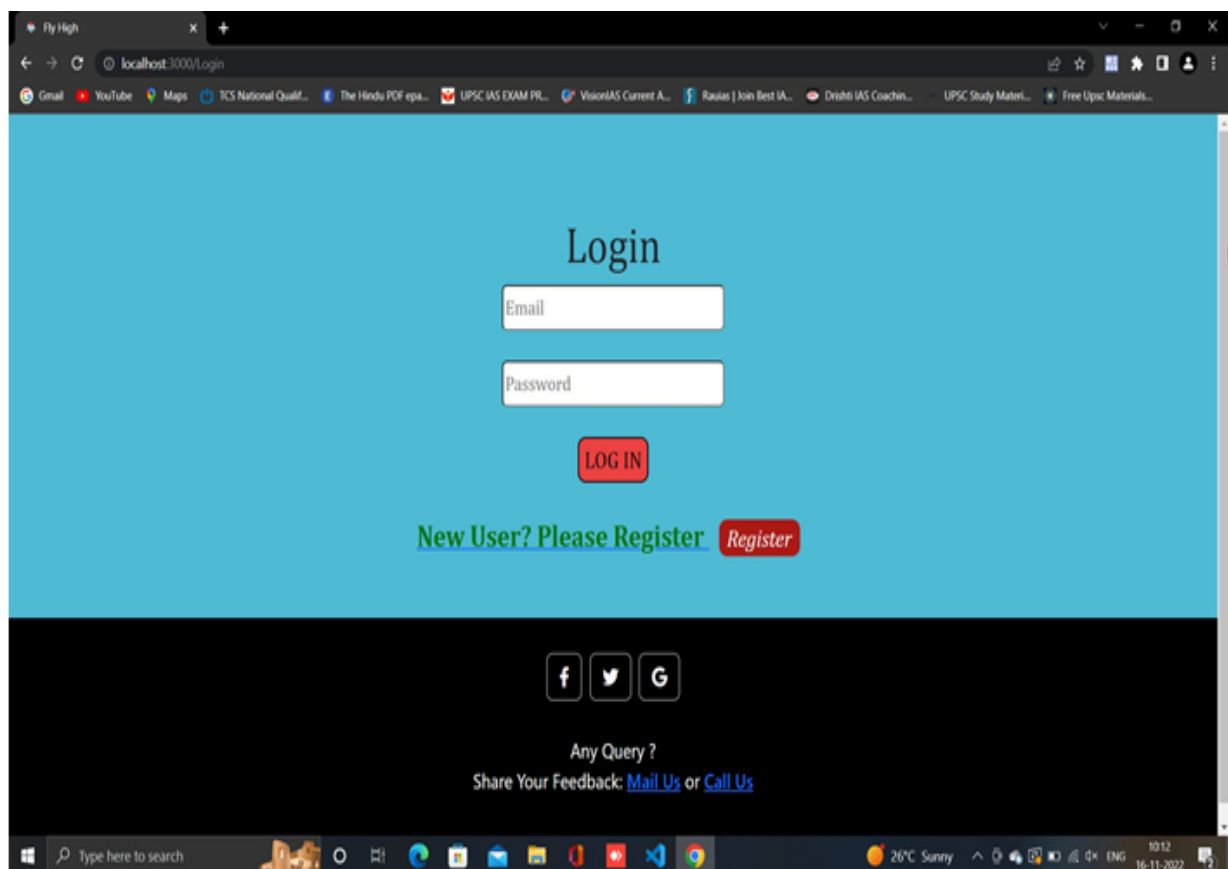




# PROJECT REPORT

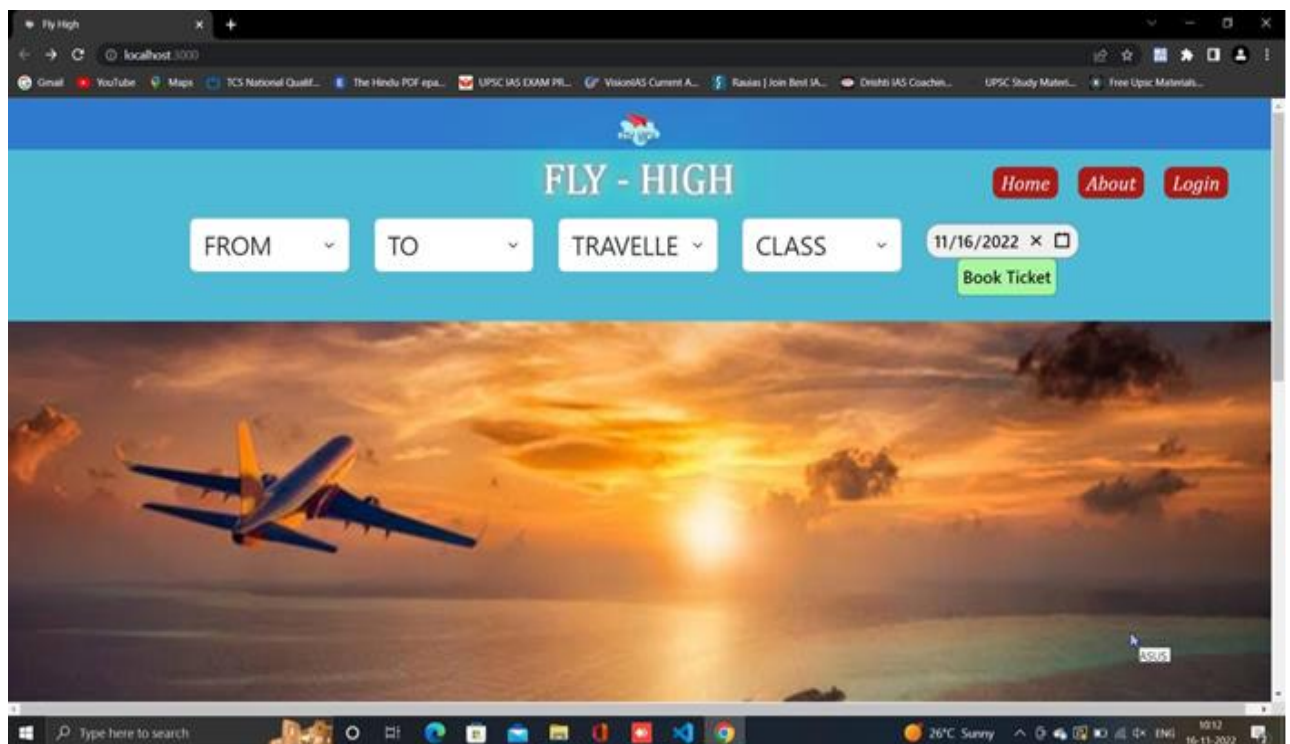


## LOG-IN PAGE: -

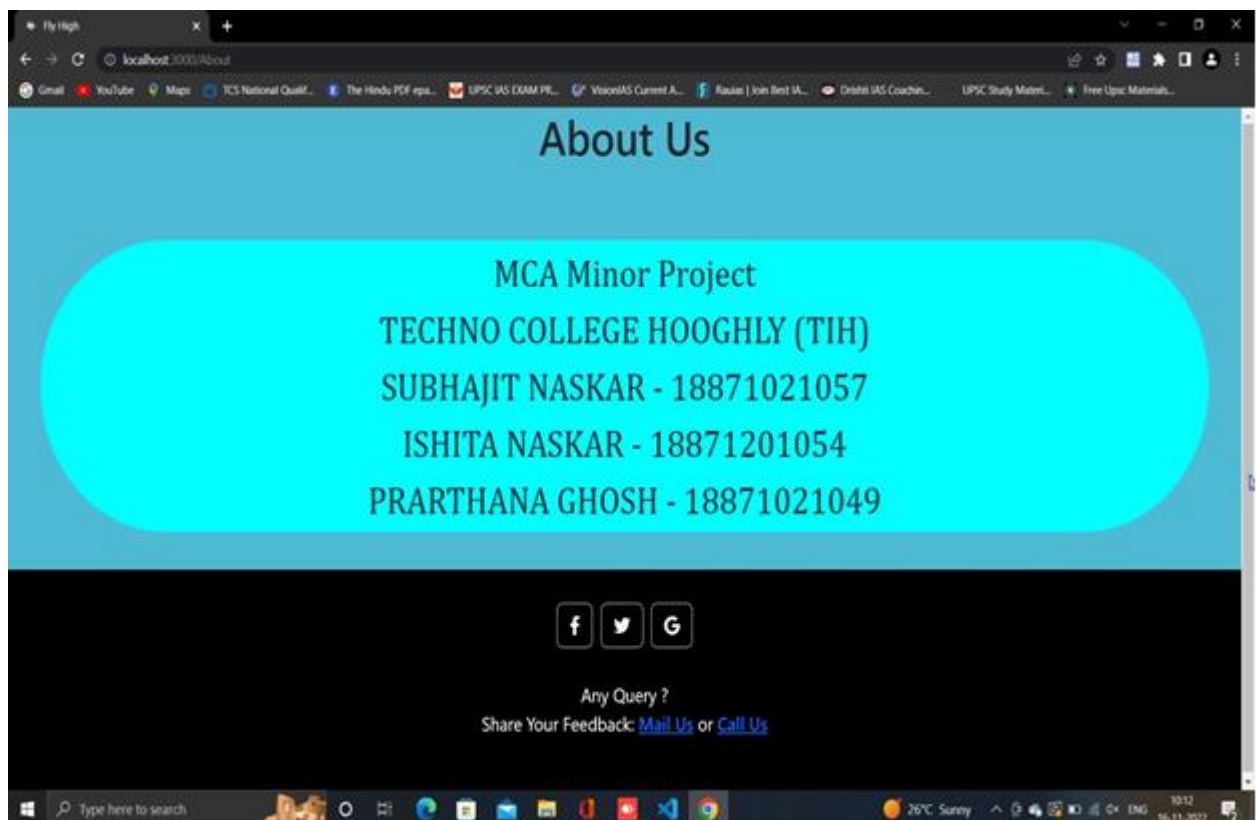


# PROJECT REPORT

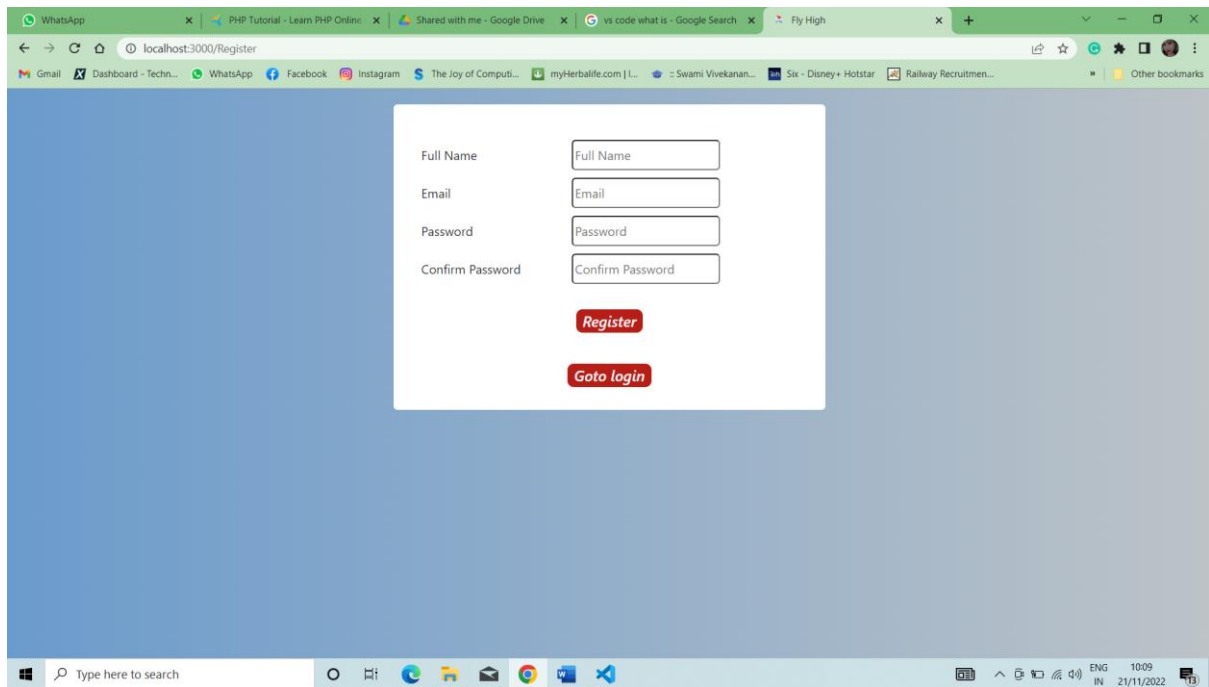
## HOME PAGE: -



## ABOUT US: -



## REGISTRATION FORM: -



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/Register'. The browser's tab bar includes 'WhatsApp', 'PHP Tutorial - Learn PHP Online', 'Shared with me - Google Drive', 'vs code what is - Google Search', and 'Fly High'. The browser's bookmark bar shows 'Gmail', 'Dashboard - Techn...', 'WhatsApp', 'Facebook', 'Instagram', 'The Joy of Comput...', 'myHerbalife.com | L...', 'Swami Vivekanan...', 'Six - Disney+ Hotstar', and 'Railway Recruitmen...'. The registration form is centered on a light blue background. It contains four input fields: 'Full Name', 'Email', 'Password', and 'Confirm Password'. Below these fields are two buttons: a red 'Register' button and a red 'Goto login' button. The Windows taskbar at the bottom shows the search bar, task view, and several application icons. The system tray on the right indicates the language is 'ENG IN', the time is '10:09', and the date is '21/11/2022'.

## 7. Testing

### Functional Testing

Functional testing is a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (Not like in white-box testing).

Functional testing differs from system testing in that functional testing a program by checking it against "... design document or specification", while system testing "validate a program by checking it against the published user or system requirements.

**Functional testing typically involves five steps:->**

1. The identification of functions that the software is expected to perform
2. The creation of input data based on the function's specifications
3. The determination of output based on the function's specifications
4. The execution of the test case
5. The comparison of actual and expected outputs

**System testing** of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

# PROJECT REPORT

## Structural Testing

It determines the durability and integrity of complete structures or sub-assemblies, usually in multi-axis testing systems that replicate end use conditions. The product range extends from single-axis component testing through to complex testing systems for the simulation of almost all service loads affecting a vehicle or component. IST offers a wide range of modular standard testing systems for use in the automotive industry to test car components such as suspensions and steering systems. Alternatively, systems can be engineered to meet specific customer requirements.

## Levels of testing

- 1) Unit testing
- 2) Integration testing
- 3) System testing
- 4) User acceptance testing

**1)unit testing:** individual software components of application are tested in isolation from other part of the program.

**big ban testing:** individual software components of an applications are combined at once into system. every module is first in it tested. Then the entire system is tested for communication interfaces between them.

**bottom-up testing:** in bottom-up integration testing begins from bottom of the module hierarchy and work up to the top to simulate higher level modules. every module is first unit tested then modules are added in ascending hierarchical order. Lower-level modules are tested first then the next set of higher-level modules are tested with previously tested lower-level modules.

**top-down testing:** begins testing from top of the module hierarchy and work down to the bottom to simulate lower interfacing modules. Every module is first unit tested then the modules are added in descending hierarchical order. Higher level modules are tested first then the next set of lower-level modules are tested with previously tested higher-level modules.

## Testing of the Project

### **Proofreading**

Proofreading (also proof-reading) traditionally means reading a proof copy of a text in order to detect and correct any errors. Modern proofreading often requires reading copy at earlier stages as well.

### **Double Entry**

In the double entry system transactions are entered twice in the accounts. For example, the same car purchased will result an increase in the "vehicle" account and a decrease in "cash"

# PROJECT REPORT

account. Therefore, the difference between the two is that in single entry system, transaction is entered only once and in double entry system it is entered twice

This is to find if any errors are present in the system. To check for the errors an artificial made database are fed to system and been checked and the errors will be verified accordingly.

**Two main ways of data verifications are: -**

**Single Method:** this is also known as proof reading method-

Typing the data twice and comparing the two databases at the same time. This method is also known as double entry method

## 8. Project Legacy

✚ Current status of project

- Data has been added
- New customer login
- Deletion in the data
- Updating
- Generate id automatically

## 9. Source code

### CODING IN REACT JS: -

#### CODING FOR LOGIN PAGE: -

```
import React,{Component} from "react";
import { NavLink } from "react-router-dom";
import './css/Login.css';
//import { Link } from "react-router-dom";
//import {useNavigate} from "react-router-dom"
//import Register from '/Register'
import Footer from './Footer';

class Login extends Component {
  constructor(props) {
    super(props)

    this.state={
      email:"",
      password:"",
    }
    this.handleSubmit=this.handleSubmit.bind(this)
  }

  emailhandler =(event) =>{
```

## PROJECT REPORT

```
    this.setState({
      email:event.target.value
    })
  }
  passwordhandler =(event) =>{
    this.setState({
      password: event.target.value
    })
  }
  handlesubmit =(event) =>{
    console.log(this.state);
    this.setState({
      email:"",
      password:"",
    })
    event.preventDefault()
  }

  render () {
    return(
      <div>
        <form onSubmit={this.handlesubmit}>
          <h1>Login</h1>
          <input type="email" value={this.state.email} onChange={this.emailhandler}
placeholder="Email"/><br/><br/>
          <input type="password" value={this.state.password}
onChange={this.passwordhandler} placeholder="Password"/><br/><br/>
          <input type="submit" value="LOG IN"/><br/><br/>
          <NavLink to="/Register" class="nav">
            <p2>New User? Please Register </p2> <h3>Register</h3>
          </NavLink>
        </form>
        <Footer/>
      </div>
    )
  }
}
```

export default Login;

### CODING FOR HOME PAGE :-

```
import React from "react";
import "../css/Home.css";
import { NavLink } from "react-router-dom";
import Header from "../Header";
import Destination from "../Destinations";
import Gallery from "../Gallery";
import Footer from "../Footer";
```

# PROJECT REPORT

```
function Home() {
  return (
    <div>
      <div className="header">
        <Header />
      </div>
      <div className="nav">
        <NavLink to="/" class="nav">
          <h3>Home</h3>
        </NavLink>
        <NavLink to="/About" class="nav">
          <h3>About</h3>
        </NavLink>
        <NavLink to="/Login" class="nav">
          <h3>Login</h3>
        </NavLink>
      </div>
      <br />
      <br />
      <div className="dropdown">
        <Destination/><br/>
      </div>
      <Gallery/>
      <Footer/>
    </div>
  );
}
export default Home;
```

## CODING FOR REGISTER PAGE :-

```
import React from "react";

function Register() {
  return(
    <div className="form">
      <div className="form-body">
        <div className="username">
          <label className="form__label" for="fullName">Full Name </label>
          <input className="form__input" type="text" id="fullName"
placeholder="Full Name"/>
        </div>
        <div className="email">
          <label className="form__label" for="email">Email </label>
          <input type="email" id="email" className="form__input"
placeholder="Email"/>
        </div>
        <div className="password">
```



## PROJECT REPORT

```
        <label className="form__label" for="password">Password </label>
        <input className="form__input" type="password" id="password"
placeholder="Password"/>
      </div>
      <div className="confirm-password">
        <label className="form__label" for="confirmPassword">Confirm
Password </label>
        <input className="form__input" type="password" id="confirmPassword"
placeholder="Confirm Password"/>
      </div>
    </div>
    <div class="footer">
      <button type="submit" class="btn">Register</button>
    </div>
  </div>
);
};
export default Register;
```

### CODING FOR COUNTER PAGE: -

```
import React, { useState } from 'react';

function Counter() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
};
export default Counter;
```

### CODING FOR DATE PICKER: -

```
import React, { useState } from 'react';
import DatePicker from 'react-date-picker';
import '../css/Datepicker.css';

function Datepicker() {
  const [value, onChange] = useState(new Date());

  return (
    <div>
      <DatePicker className='date' onChange={onChange} value={value} />
      <div class="btn-group" role="group" aria-label="Basic mixed styles example">
```



## PROJECT REPORT

```
<a href='Counter'><button type="button" class="btn btn-outline-success">Book
Ticket</button></a>
</div>
</div>
);
};
export default Datepicker;
```

### CODING FOR DESTINATION: -

```
import React from "react";
import { Row } from "react-bootstrap";
import { Col } from "react-bootstrap";
import Datepicker from "../Datepicker";
import '../css/Destination.css';

function Destination() {
  return (
    <div>
      <Row><Col></Col>
      <Col>
        <select class="form-select" aria-label="Default select example">
          <option selected>FROM</option>
          <option value="AJL"> Aizawl Airport</option>
          <option value="IXU"> Aurangabad Airport</option>
          <option value="ATQ"> Amritsar International Airport</option>
          <option value="IXB"> Bagdogra International Airport</option>
          <option value="IXG"> Belagavi International Airport</option>
          <option value="BLR"> Bengaluru International Airport</option>
          <option value="BBI"> Bhubaneswar International Airport</option>
          <option value="BHO"> Bhopal International Airport</option>
          <option value="IXC"> Chandigarh International Airport</option>
          <option value="MAA"> Chennai International Airport</option>
          <option value="CJB"> Coimbatore International Airport</option>
          <option value="DED"> Dehradun International Airport</option>
          <option value="DEL"> Delhi International Airport</option>
          <option value="DIB"> Dibrugarh Airport</option>
          <option value="DMU"> Dimapur International Airport</option>
          <option value="RDP"> Durgapur Airport</option>
          <option value="GAY"> Gaya International Airport</option>
          <option value="GOY"> Goa International Airport</option>
          <option value="GOP"> Gorakhpur International Airport</option>
          <option value="GAU"> Guwahati International Airport</option>
          <option value="HBX"> Hubli International Airport</option>
          <option value="HYD"> Hyderabad International Airport</option>
          <option value="IMF"> Imphal International Airport</option>
          <option value="IDR"> Indore International Airport</option>
          <option value="JLR"> Jabalpur International Airport</option>
          <option value="JAI"> Jaipur International Airport</option>
          <option value="IXJ"> Jammu International Airport</option>
```

## PROJECT REPORT

```
<option value="JDH"> Jodhpur International Airport</option>
<option value="JRH"> Jorhat Airport</option>
<option value="CNN"> Kannur International Airport</option>
<option value="KJB"> Kurnool Airport</option>
<option value="COK"> Kochi Airport</option>
<option value="KLH"> Kolhapur Airport</option>
<option value="CCU"> Kolkata International Airport</option>
<option value="CCJ"> Kozhikode International Airport</option>
<option value="LKO"> Lucknow International Airport</option>
<option value="IXM"> Madurai International Airport</option>
<option value="IXE"> Mangaluru International Airport</option>
<option value="BOM"> Mumbai International Airport</option>
<option value="MYQ"> Moiseru Airport</option>
<option value="NAG"> Nagpur International Airport</option>
<option value="PAT"> Patna International Airport</option>
<option value="IXD"> Prayagraj Airport</option>
<option value="PNQ"> Pune International Airport</option>
<option value="IXZ"> Portblair Airport</option>
<option value="RPR"> Raipur International Airport</option>
<option value="RJA"> Rajahmundry International Airport</option>
<option value="IXR"> Ranchi International Airport</option>
<option value="SHL"> Shillong Airport</option>
<option value="SAG"> Shirdi Airport</option>
<option value="IXS"> Silchar International Airport</option>
<option value="SXR"> Srinagar International Airport</option>
<option value="STV"> Surat International Airport</option>
<option value="TRV"> Tiruvananthapuram International
```

Airport</option>

```
<option value="TRZ"> Tiruchirappalli International Airport</option>
<option value="TIR"> Tirupati Airport</option>
<option value="TCR"> Tuticorin International Airport</option>
<option value="UDR"> Udaipur International Airport</option>
<option value="BDQ"> Vadodara International Airport</option>
<option value="VNS"> Varanasi International Airport</option>
<option value="VGA"> Vijaywada International Airport</option>
<option value="VTZ"> Visakhapatnam International Airport</option>
</select></Col>
```

<Col>

```
<select class="form-select" aria-label="Default select example">
  <option selected>TO</option>
  <option value="AJL"> Aizawl Airport</option>
  <option value="IXU"> Aurangabad Airport</option>
  <option value="ATQ"> Amritsar International Airport</option>
  <option value="IXB"> Bagdogra International Airport</option>
  <option value="IXG"> Belagavi International Airport</option>
  <option value="BLR"> Bengaluru International Airport</option>
  <option value="BBi"> Bhubaneswar International Airport</option>
  <option value="BHO"> Bhopal International Airport</option>
  <option value="IXC"> Chandigarh International Airport</option>
```

## PROJECT REPORT

<option value="MAA"> Chennai International Airport</option>  
<option value="CJB"> Coimbatore International Airport</option>  
<option value="DED"> Dehradun International Airport</option>  
<option value="DEL"> Delhi International Airport</option>  
<option value="DIB"> Dibrugarh Airport</option>  
<option value="DMU"> Dimapur International Airport</option>  
<option value="RDP"> Durgapur Airport</option>  
<option value="GAY"> Gaya International Airport</option>  
<option value="GOY"> Goa International Airport</option>  
<option value="GOP"> Gorakhpur International Airport</option>  
<option value="GAU"> Guwahati International Airport</option>  
<option value="HBX"> Hubli International Airport</option>  
<option value="HYD"> Hyderabad International Airport</option>  
<option value="IMF"> Imphal International Airport</option>  
<option value="IDR"> Indore International Airport</option>  
<option value="JLR"> Jabalpur International Airport</option>  
<option value="JAI"> Jaipur International Airport</option>  
<option value="IXJ"> Jammu International Airport</option>  
<option value="JDH"> Jodhpur International Airport</option>  
<option value="JRH"> Jorhat Airport</option>  
<option value="CNN"> Kannur International Airport</option>  
<option value="KJB"> Kurnool Airport</option>  
<option value="COK"> Kochi Airport</option>  
<option value="KLH"> Kolhapur Airport</option>  
<option value="CCU"> Kolkata International Airport</option>  
<option value="CCJ"> Kozhikode International Airport</option>  
<option value="LKO"> Lucknow International Airport</option>  
<option value="IXM"> Madurai International Airport</option>  
<option value="IXE"> Mangaluru International Airport</option>  
<option value="BOM"> Mumbai International Airport</option>  
<option value="MYQ"> Moiseru Airport</option>  
<option value="NAG"> Nagpur International Airport</option>  
<option value="PAT"> Patna International Airport</option>  
<option value="IXD"> Prayagraj Airport</option>  
<option value="PNQ"> Pune International Airport</option>  
<option value="IXZ"> Portblair Airport</option>  
<option value="RPR"> Raipur International Airport</option>  
<option value="RJA"> Rajahmundry International Airport</option>  
<option value="IXR"> Ranchi International Airport</option>  
<option value="SHL"> Shillong Airport</option>  
<option value="SAG"> Shirdi Airport</option>  
<option value="IXS"> Silchar International Airport</option>  
<option value="SXR"> Srinagar International Airport</option>  
<option value="STV"> Surat International Airport</option>  
<option value="TRV"> Tiruvananthapuram International  
Airport</option>  
<option value="TRZ"> Tiruchirappalli International Airport</option>  
<option value="TIR"> Tirupati Airport</option>  
<option value="TCR"> Tuticorin International Airport</option>  
<option value="UDR"> Udaipur International Airport</option>

## PROJECT REPORT

```
        <option value="BDQ"> Vadodara International Airport</option>
        <option value="VNS"> Varanasi International Airport</option>
        <option value="VGA"> Vijaywada International Airport</option>
        <option value="VTZ"> Visakhapatnam International Airport</option>
    </select></Col>
</Col>
<select class="form-select" aria-label="Default select example">
    <option selected>TRAVELLER</option>
    <option value="1"> 1</option>
    <option value="2"> 2</option>
    <option value="3"> 3</option>
    <option value="4"> 4</option>
    <option value="5"> 5</option>
    <option value="6"> 6</option>
    <option value="7"> 7</option>
    <option value="8"> 8</option>
    <option value="9"> 9</option>
    <option value="9+"> 9+</option>
</select></Col>
</Col>
<select class="form-select" aria-label="Default select example">
    <option selected>CLASS</option>
    <option value="1"> Economy</option>
    <option value="2"> Business</option>
    <option value="3"> Premium Economy</option>
</select></Col>
<Col><Datepicker/></Col>
<Col></Col>
</Row>
</div>
);
};
export default Destination;
CODING FOR HEADER: -
import React from "react";
import logo from "../images/logo1.png";
import "../css/Header.css";

function Header() {
    return (
        <div class="cotainer-fluid ">
            <nav>
                <div class="container">
                    <img src={logo} alt="logo" />
                    <h1 class="neonText">FLY - HIGH</h1>
                </div>
            </nav>
        </div>
    );
}
```

## PROJECT REPORT

export default Header;

### CODING FOR FOOTER: -

```
import React from "react";
```

```
function Footer() {
```

```
  return (
```

```
    <footer className="bg-black text-center text-white">
```

```
      <div className="container p-4">
```

```
        <section className="mb-4">
```

```
          <a className="btn btn-outline-light btn-floating m-1" href="/"
role="button"><i
```

```
            className="fab fa-facebook-f"></i></a>
```

```
          <a className="btn btn-outline-light btn-floating m-1" href="/"
role="button"><i className="fab fa-twitter"></i></a>
```

```
          <a className="btn btn-outline-light btn-floating m-1" href="/"
role="button"><i className="fab fa-google"></i></a>
```

```
        </section>
```

```
      <section className="mb-4">
```

```
        <p>
```

```
        Any Query ? <br/>
```

```
        Share Your Feedback: <a href="mailto:s.n240819@gmail.com">Mail
Us</a> or <a href="tel:6291338577">Call Us</a> <br/>
```

```
        </p>
```

```
      </section>
```

```
    </div>
```

```
  </footer>
```

```
);
```

```
};
```

```
export default Footer;
```

### CODING FOR GALLERY: -

```
import React from "react";
```

```
import img1 from '../images/flight1.jpg';
```

```
import img2 from '../images/flight2.jpg';
```

```
import img3 from '../images/flight3.jpg';
```

```
import img4 from '../images/flight4.jpg';
```

```
import img5 from '../images/flight5.jpg';
```

```
import img6 from '../images/flight6.jpg';
```

```
import '../css/Gallery.css';
```

```
import {
```

```
  MDBCarousel,
```

```
  MDBCarouselItem,
```

```
} from 'mdb-react-ui-kit';
```

# PROJECT REPORT

```
export default function App() {
  return (
    <MDBCarousel showControls showIndicators>
      <MDBCarouselItem
        className='w-100 d-block'
        itemId={1}
        src={img1}
        alt='...'
      />
      <MDBCarouselItem
        className='w-100 d-block'
        itemId={2}
        src={img2}
        alt='...'
      />
      <MDBCarouselItem
        className='w-100 d-block'
        itemId={3}
        src={img3}
        alt='...'
      />
      <MDBCarouselItem
        className='w-100 d-block'
        itemId={4}
        src={img4}
        alt='...'
      />
      <MDBCarouselItem
        className='w-100 d-block'
        itemId={5}
        src={img5}
        alt='...'
      />
      <MDBCarouselItem
        className='w-100 d-block'
        itemId={6}
        src={img6}
        alt='...'
      />
    </MDBCarousel>
  );
}
```

## CODING FOR ABOUT: -

```
import React from "react";
import Footer from './Footer';
import '../css/About.css';
```

```
function About(){
  return(
    <div>
```

# PROJECT REPORT

```
<h1>About Us</h1><br/>
<div className="details">
MCA Minor Project<br/>
TECHNO COLLEGE HOOGHLY (TIH)<br/>
SUBHAJIT NASKAR - 18871021057<br/>
ISHITA NASKAR - 18871201054<br/>
PRARTHANA GHOSH - 18871021049
</div>
<div>
  <Footer/>
```

```
    </div>
  </div>
);
};
export default About;
```

## CODING FOR APP:-

```
import './App.css';
import {BrowserRouter, Routes, Route} from 'react-router-dom';
import Home from './component/Home';
import About from './component/About';
import Login from './component/Login';
```

```
function App() {
  return (
    <div className="App">
      <BrowserRouter>
        <Routes>
          <Route exact path="/" element={<Home/>}>Home</Route>
          <Route exact path="/About" element={<About/>}>About</Route>
          <Route exact path="/Login" element={<Login/>}>Login</Route>
        </Routes>
      </BrowserRouter>
      <div className="text-wrapper">
        <div>
          </div>
        </div>
      </div>
    );
  }
}
```

```
export default App;
```

## CODING FOR INDEX:-

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
```

# PROJECT REPORT

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

## CODING FOR CSS

### CODING FOR LOGIN PAGE : -

```
cursor: pointer;
color: #7439db;
}

.link-btn {
  background: none;
  color: white;
  text-decoration: underline;
} */

h1{
  text-align: center;
}
h1:hover{
  color: rgb(6, 85, 105);
}
body{
  margin: 0;
  background-color: rgb(79, 186, 212);
}
form{
  font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
  border-radius: 1rem;
  margin-top: 2rem!important;
  margin:0 auto;
  padding:3rem;
}
input{
  border-radius: .3rem;
```



# PROJECT REPORT

```
    line-height: 2rem;
}
/*
input[type="text"]{
    width:90%;
    border:.1rem solid;
    padding:1.4rem;
    border-radius:1rem;
    font-size: 1rem;
}
input[type="password"]{
    background: #1d1d1d;
    border-radius:2rem;
    border:.3rem solid;
} */
input[type="submit"]{
    background: rgb(235, 66, 66);
    border-radius: .5rem;
    font-size: 1.2rem;
}
p2{
    color:rgb(4, 117, 10) ;
    font-size: 1.5rem;
}
a{
    font-weight: 600;
}
}
```

## CODING FOR HOME PAGE : -

```
.nav {
    float: right;
    margin-right: 2rem;
    font-family: Cambria, Cochin, Georgia, Times, "Times New Roman", serif;
}
h3 {
    background-color: rgba(179, 15, 9, 0.938);
    color: rgb(244, 245, 247);
    display: inline-block;
    font-size: 1.2rem;
    margin: 0.6rem;
    margin-block-start: 0;
    margin-block-end: 0.1rem;
    padding: 0.2rem 0.5rem;
    border-radius: 0.5rem;
    font-style: oblique;
}
h3:hover {
    transform: scale(1.15);
}
p1 {
    font-size: 2.5rem;
}
```

## PROJECT REPORT

```
background-color: rgb(177, 226, 210);
padding: 0.3rem 0.8rem;
border-radius: 1rem;
font-weight: 600;
font-style: oblique;
}
```

### CODING FOR HEADER: -

```
nav{
  font-family: Cambria, Cochin, Georgia, Times, "Times New Roman", serif;
  margin-top: 0rem;
  margin-bottom: 1rem;
  background-color: rgba(19, 66, 196, 0.534);
  color: rgb(51, 179, 83);
  font-size: 2.1rem;
  height: 3rem;
  font-weight: 600;
}
img{
  width: 0.1;
  height: 0.1;
}
h1{
  font-size: 2.5rem;
}

.neonText {
  animation: flicker 1.5s infinite alternate;
  color: #fff;
  text-shadow: 0 0 4px rgb(107, 180, 230), 0 0 11px rgb(202, 68, 68), 0 0 19px #fff, 0 0
40px #0fa,
    0 0 80px rgb(0, 140, 255), 0 0 90px #0fa, 0 0 100px #0fa, 0 0 150px #0fa;
}
```

### CODING FOR COUNTER: -

```
.control_btn:hover {
  background-color: rgb(16, 0, 54);
  color: rgb(255, 255, 255);
  border: 1px solid rgb(16, 0, 54);
}

.reset {
  width: 100px;
  font-size: 18px;
  padding: 10px 20px;
  background-color: transparent;
  color: rgb(16, 0, 54);
  border: 1px solid rgb(16, 0, 54);
  cursor: pointer;
  transition: 0.2s ease-in-out;
}
```

# PROJECT REPORT

```
.reset:hover {  
    background-color: rgb(16, 0, 54);  
    color: rgb(255, 255, 255);  
    border: 1px solid rgb(16, 0, 54);  
}
```

```
.counter__output {  
    font-size: 40px;  
    color: rgb(116, 7, 7);  
}
```

```
.btn__container {  
    display: flex;  
    justify-content: center;  
    flex-direction: row;  
    column-gap: 20px;  
}
```

```
.control__btn {  
    font-size: 20px;  
    padding: 10px 20px;  
    background-color: transparent;  
    color: rgb(16, 0, 54);  
    border: 1px solid rgb(16, 0, 54);  
    cursor: pointer;  
    transition: 0.2s ease-in-out;  
}
```

```
.counter{  
    width: 100%;  
    display: flex;  
    align-items: center;  
    flex-direction: column;  
    text-align: center;  
    row-gap: 20px;  
}
```

```
.counter h1 {  
    color: rgb(16, 0, 54);  
    font-size: 40px;  
    font-family: cursive;  
}
```

## CODING FOR DATEPICKER: -

```
/* .date{  
    background-color: aqua;  
    margin-right: 1rem;  
    border-radius: 1rem;  
    border: none;  
} */
```

# PROJECT REPORT

```
.react-date-picker__wrapper{
  margin-top: .3rem;
  margin-right: .5rem;
  background-color: rgb(240, 240, 240);
  border-radius: .8rem;
  padding: .2rem;
  font-weight: 500;
}
.btn-outline-success{
  font: 1rem;
  margin-top: 0;
  background-color:rgb(171, 246, 162);
  color: black;
  font-weight: 500;
  padding: .3rem;
}
```

## CODING FOR DESTINATION: -

```
.form-select{
  font-size: 1.5rem;
}
```

## CODING FOR GALLERY: -

```
.container h2{
  font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
  font-size: 2rem ;
  font-weight: 500;
  color: rgb(6, 33, 74);
}
.gallery{
  border-radius: 1rem;
  padding: .2rem;
}
```

## CODING FOR ABOUT: -

```
.details{
  background-color: aqua;
  font-size: 2rem;
  margin: 2rem;
  color: rgb(2, 21, 37);
  font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
  padding: .200rem;
  border-radius: 20rem;
}
```

## CODING FOR REGISTRATION: -

```
body{
  background: #bdc3c7; /* fallback for old browsers */
  background: -webkit-linear-gradient(to right, #2c3e50, #bdc3c7); /* Chrome 10-25,
Safari 5.1-6 */
  background: linear-gradient(to right, #2c3e50, #bdc3c7); /* W3C, IE 10+/ Edge,
Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */
}
```

# PROJECT REPORT

```
}

.form{
  background-color: white;
  border-radius: 5px;
  width: 550px;
  margin: 20px auto;
  padding: 20px;
  /* height: 600px; */
}

.form-body{
  text-align: left;
  padding: 20px 10px;
}

.form-body > *{
  padding: 5px;
}

.form__label{
  width: 40%;
}

.form_input{
  width: 60%;
}

.footer{
  text-align: center;
}
```

## CODING FOR APP :-

```
.App {
  text-align: center;
}

.Header {
  text-align: center;
}

/* .App .text-wrapper h1{
  color: white;
  display: inline-block;
  background-color: tomato;
  padding: 3rem 6rem;
} */

.App .text-wrapper p {
```

# PROJECT REPORT

```
color: rgb(255, 253, 253);
font-weight: 600;
width: 100vw;
height: 100vh;
margin-left: auto;
margin-right: auto;
font-size: 22px;
text-align: left;
}

.App video {
background-size: cover;
background-repeat: no-repeat;
position: absolute;
z-index: -1;
top: 0;
left: 0;
width: 100%;
height: auto;
}
@media (max-width:526px){
html{
font-size: 50%;
background-image: url('./images/Nature.jpg');
}
}
```

## 10. Bibliography

<http://www.dreamincode.net/forums/topic/53926-online-examination-system-on-vbnet/>

<http://www.seminarprojects.com/Thread-online-examination-project>