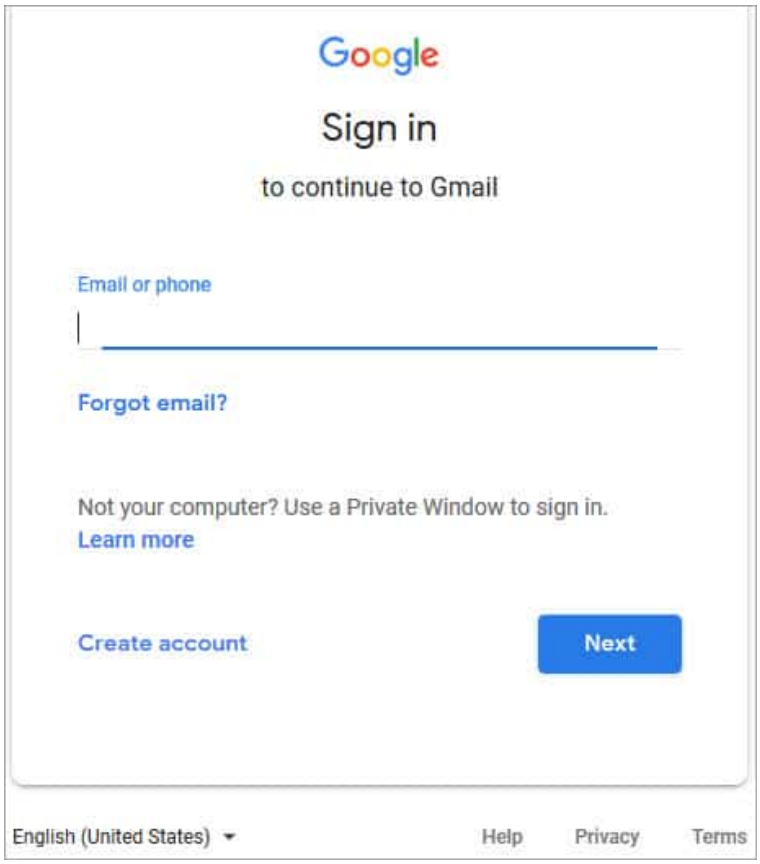
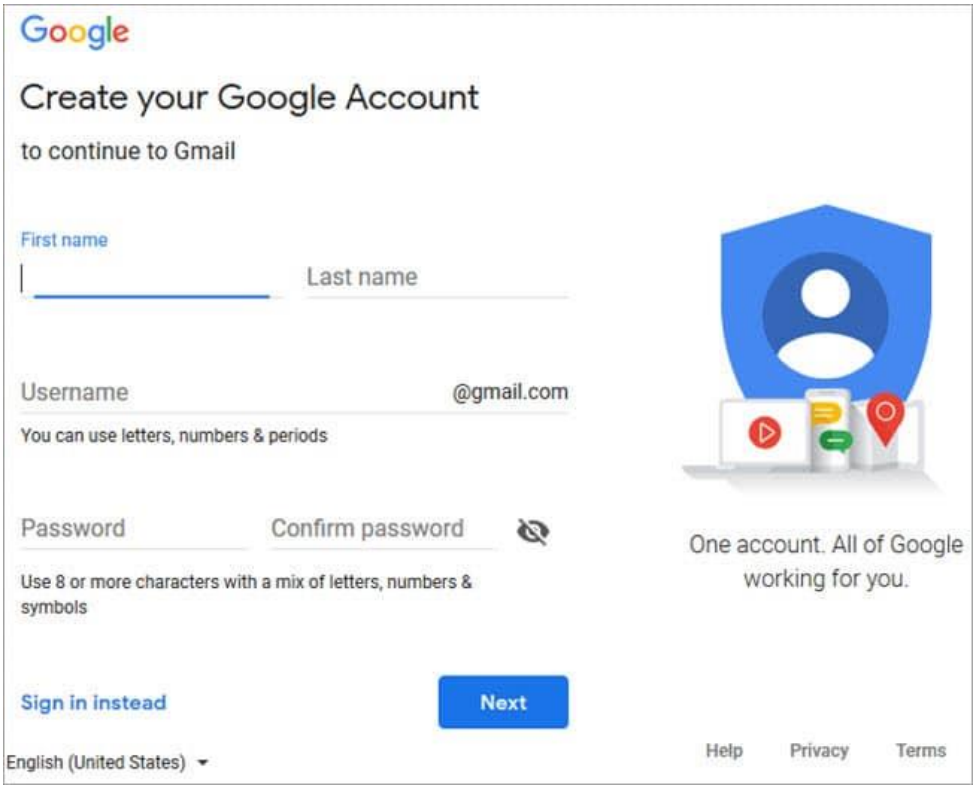
Assignment No 1

Title: Write test scenario for Gmail Login page.





Sr. No.	Test Scenarios
1	Enter the valid email address & click next. Verify if the user gets an option to enter the password.
2	Don't enter an email address or phone number & just click the Next button. Verify if the user will get the correct message or if the blank field will get highlighted.
3	Enter the invalid email address & click the Next button. Verify if the user will get the correct message.
4	Enter an invalid phone number & click the Next button. Verify if the user will get the correct message.
5	Verify if a user can log in with a valid email address and password.
6	Verify if a user can log in with a valid phone number and password.
7	Verify if a user cannot log in with a valid phone number and an invalid password.
8	Verify if a user cannot log in with a valid email address and a wrong password.
9	Verify the 'Forgot email' functionality.
10	Verify the 'Forgot password' functionality.

Test Scenarios for the Sign-up page

#1) Verify the messages for each mandatory field.

#2) Verify if the user cannot proceed without filling all the mandatory fields.

#3) Verify the age of the user when the DOB is selected.

#4) Verify if the numbers and special characters are not allowed in the First and Last name.

#5) Verify if a user can sign-up successfully with all the mandatory details.

#6) Verify if a user can log in with the valid details.

#7) Verify if the Password and Confirm Password fields are accepting similar strings only.

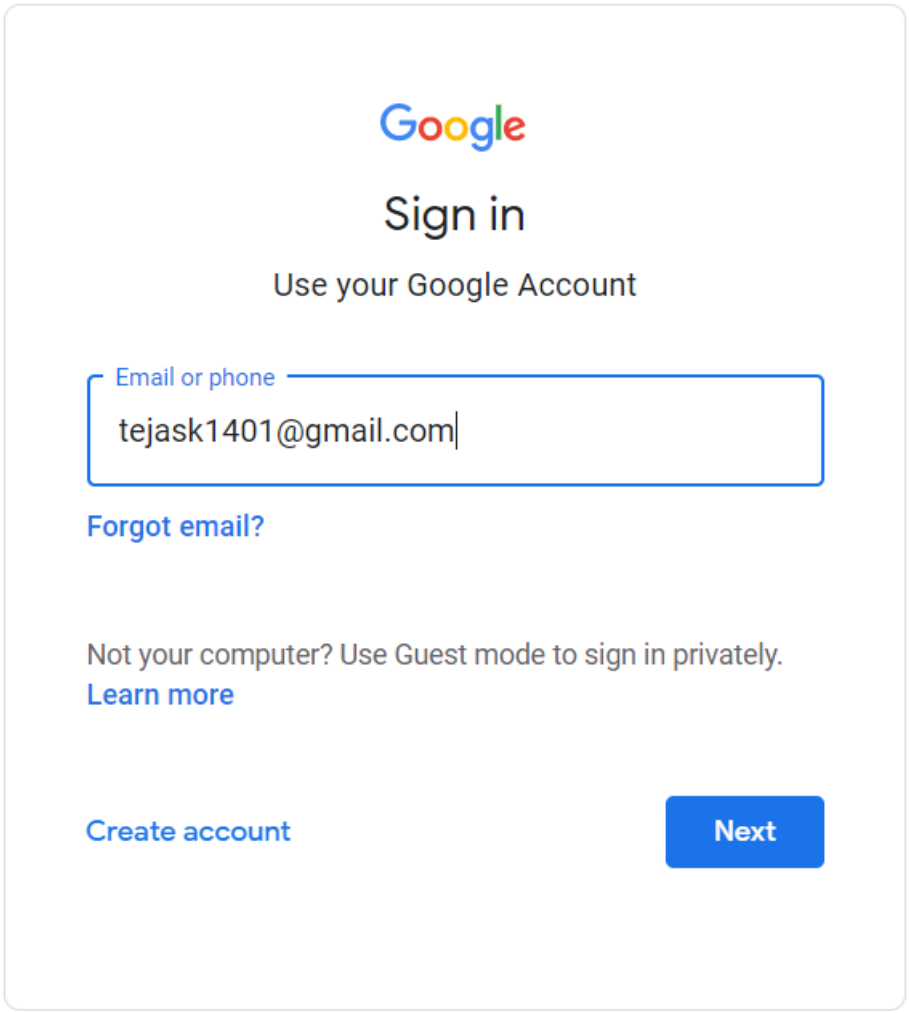
#8) Verify if the Password field will prompt you for the weak passwords.

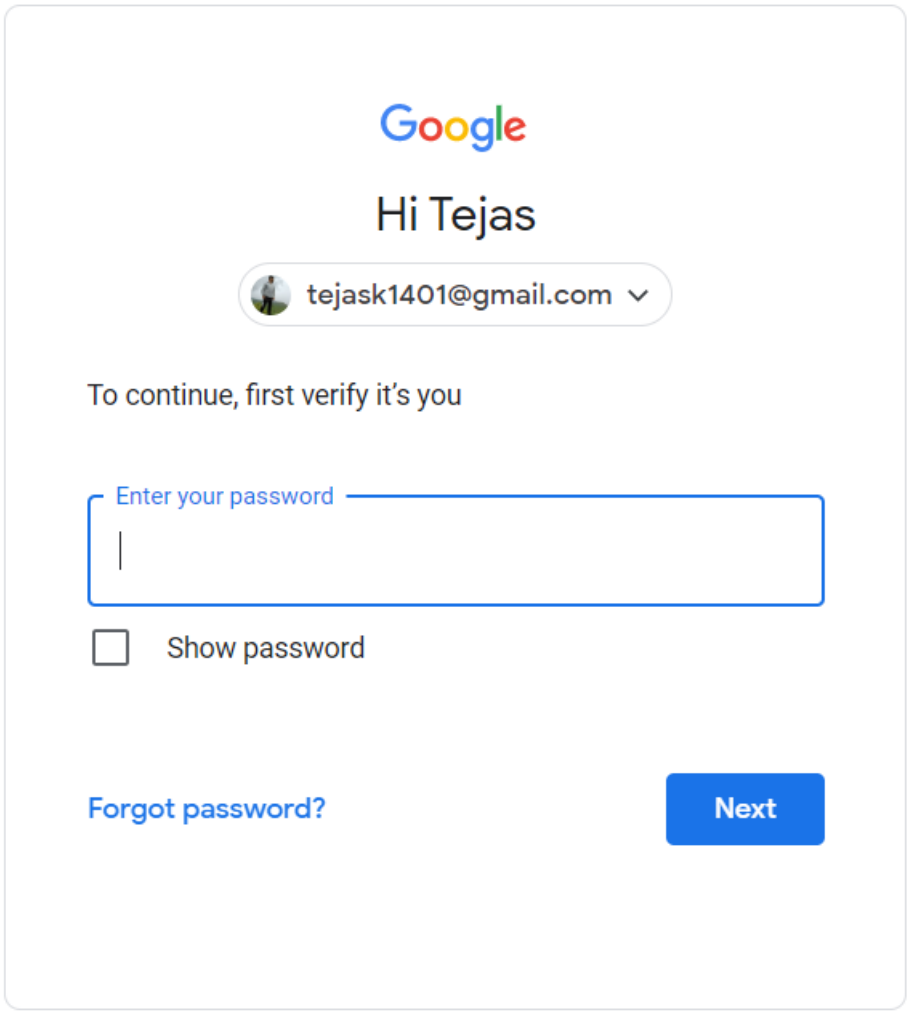
#9) Verify if duplicate email address will not get assigned.

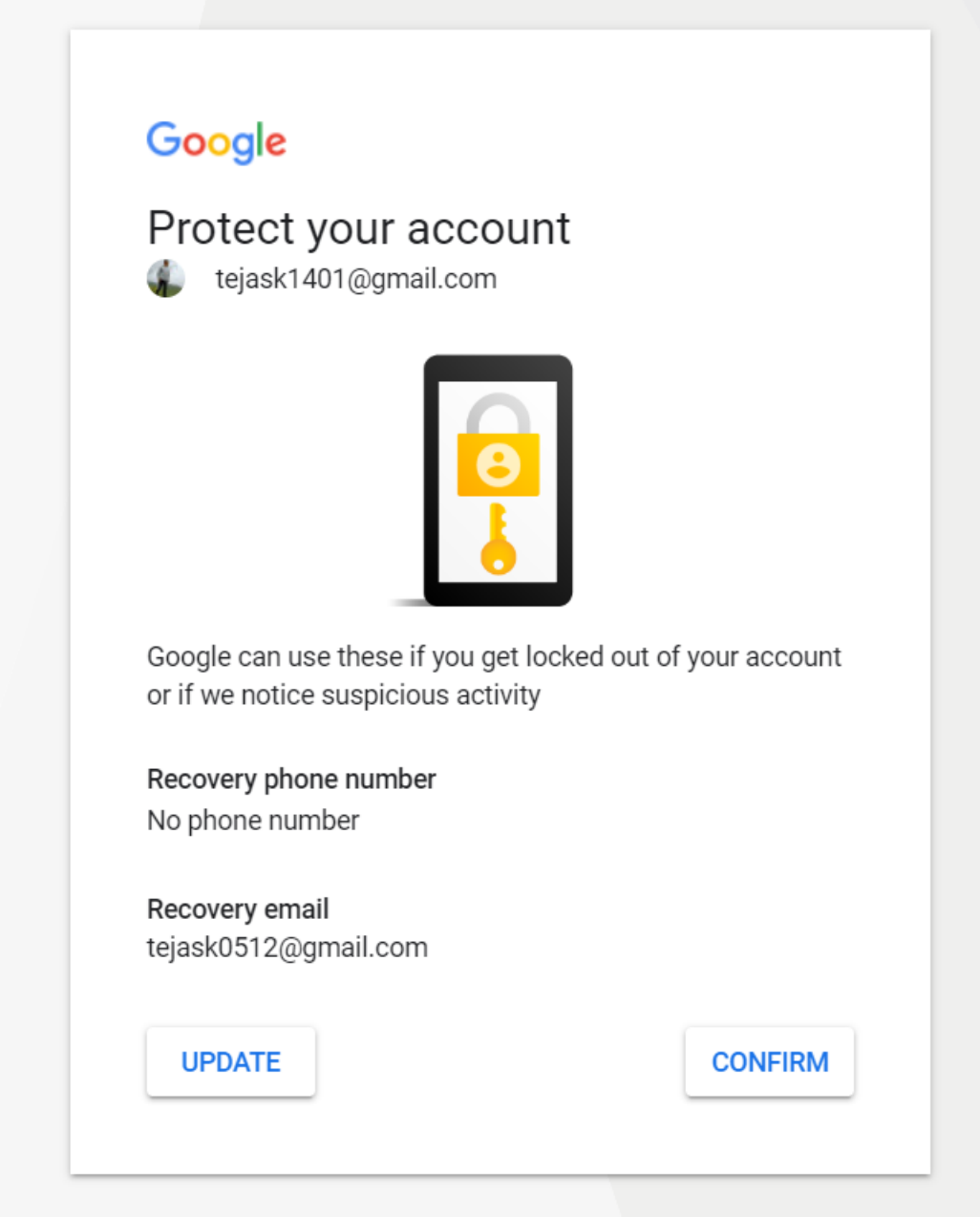
#10) Verify that hints are provided for each field on the form, for the ease of use.

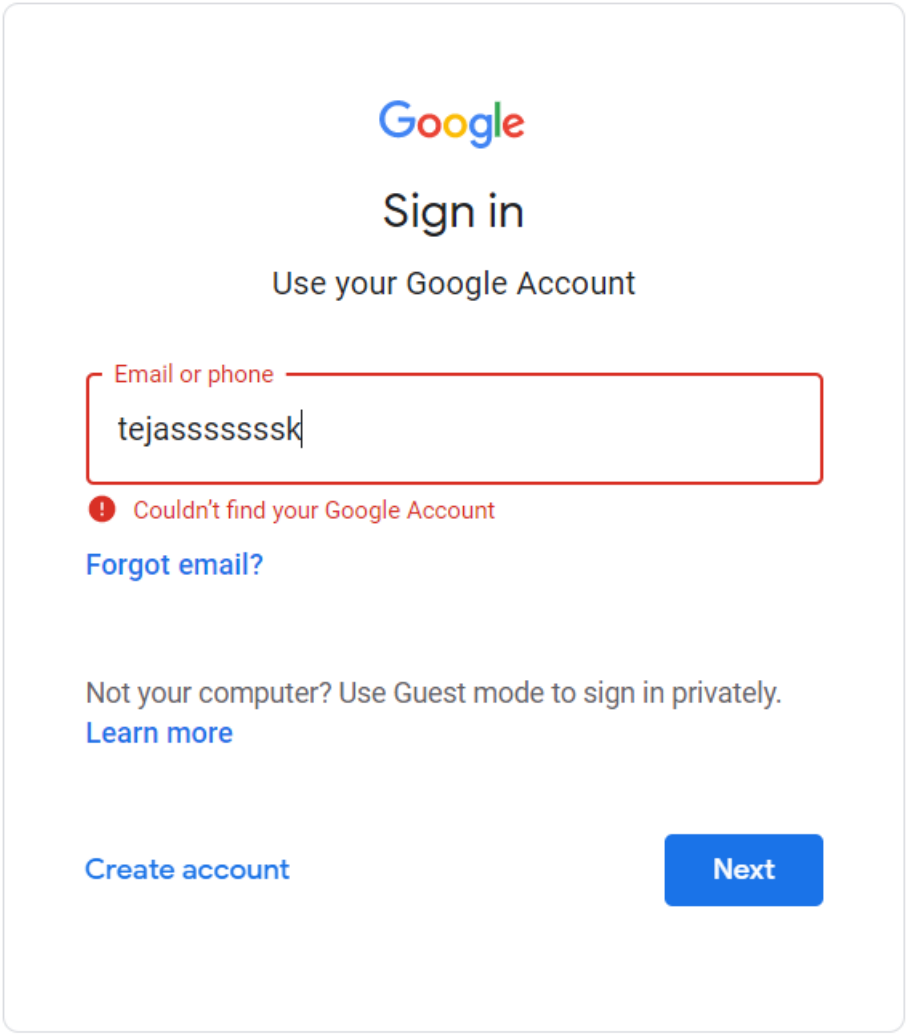
Assignment No 2

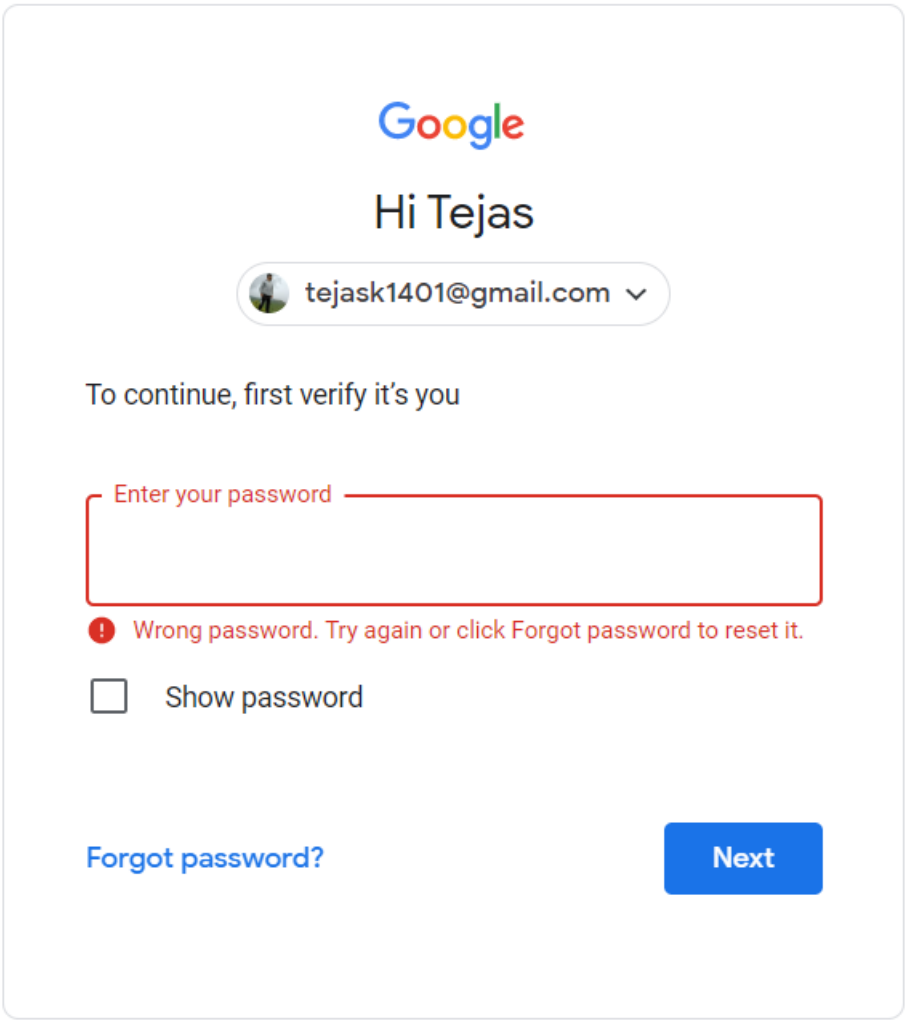
Title: TEST Scenario For Gmail Login Page.

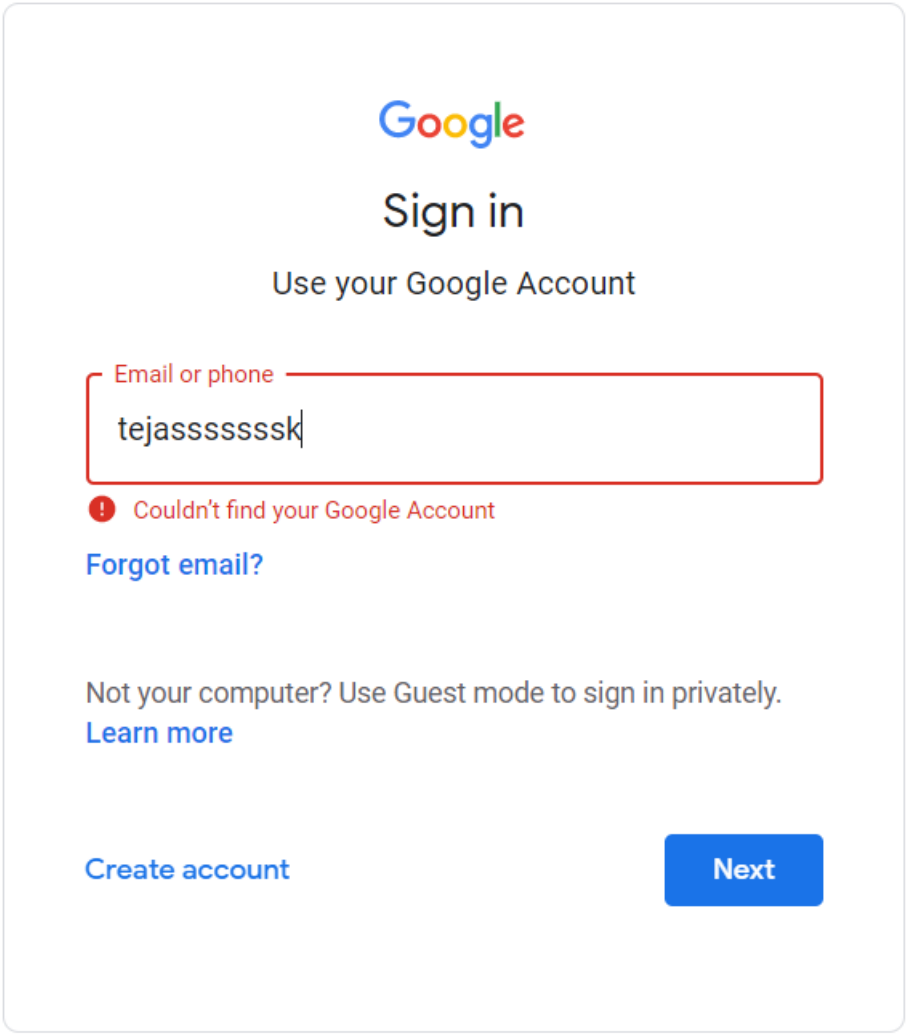



Google
Hi Tejas
tejask1401@gmail.com
To continue, first verify it's you
Enter your password
Show password
Forgot password?
Next


Google
Protect your account
tejask1401@gmail.com
Google can use these if you get locked out of your account or if we notice suspicious activity
Recovery phone number
No phone number
Recovery email
tejask0512@gmail.com
UPDATE
CONFIRM

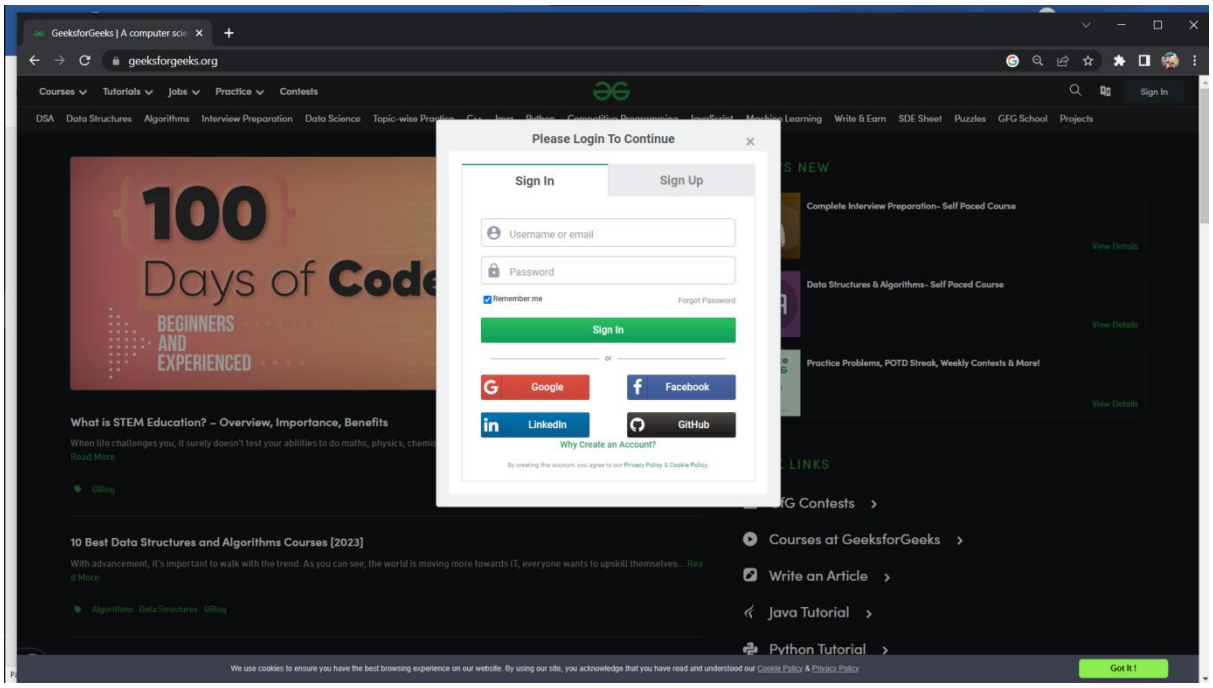

Google
Sign in
Use your Google Account
Email or phone
tejasssssssk
Couldn't find your Google Account
Forgot email?
Not your computer? Use Guest mode to sign in privately.
Learn more
Create account
Next

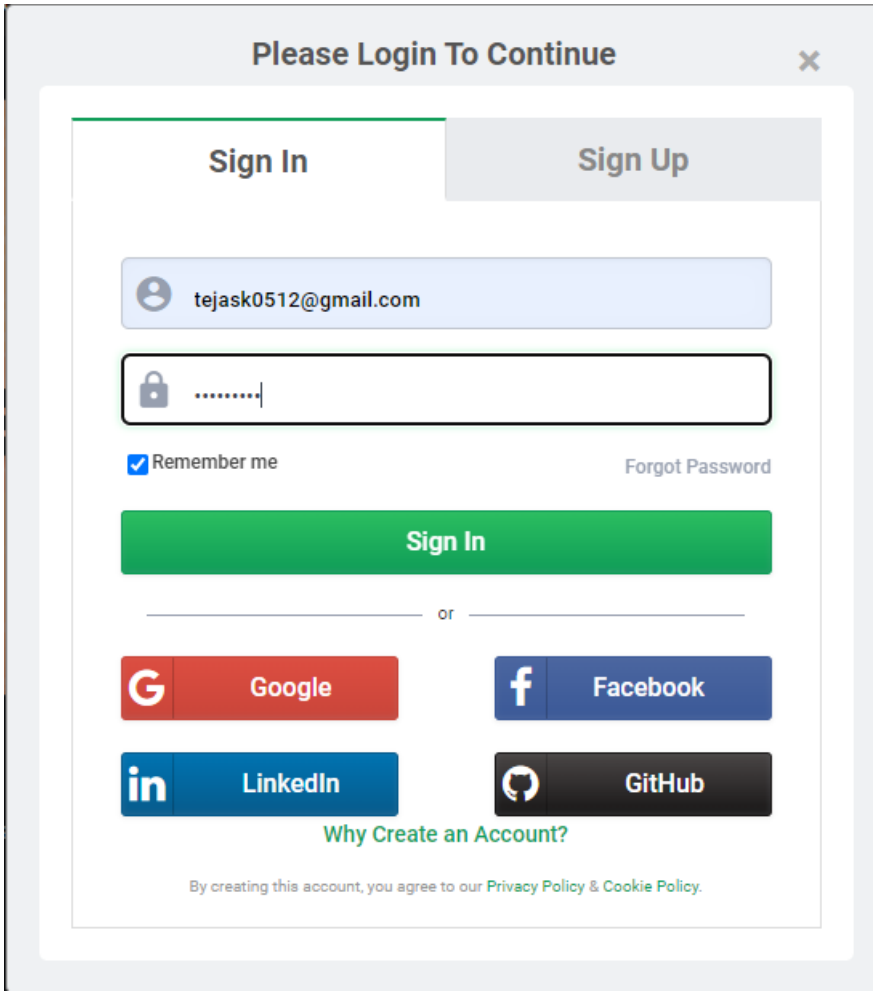

Google
Hi Tejas
tejask1401@gmail.com
To continue, first verify it's you
Enter your password
Wrong password. Try again or click Forgot password to reset it.
Show password
Forgot password?
Next

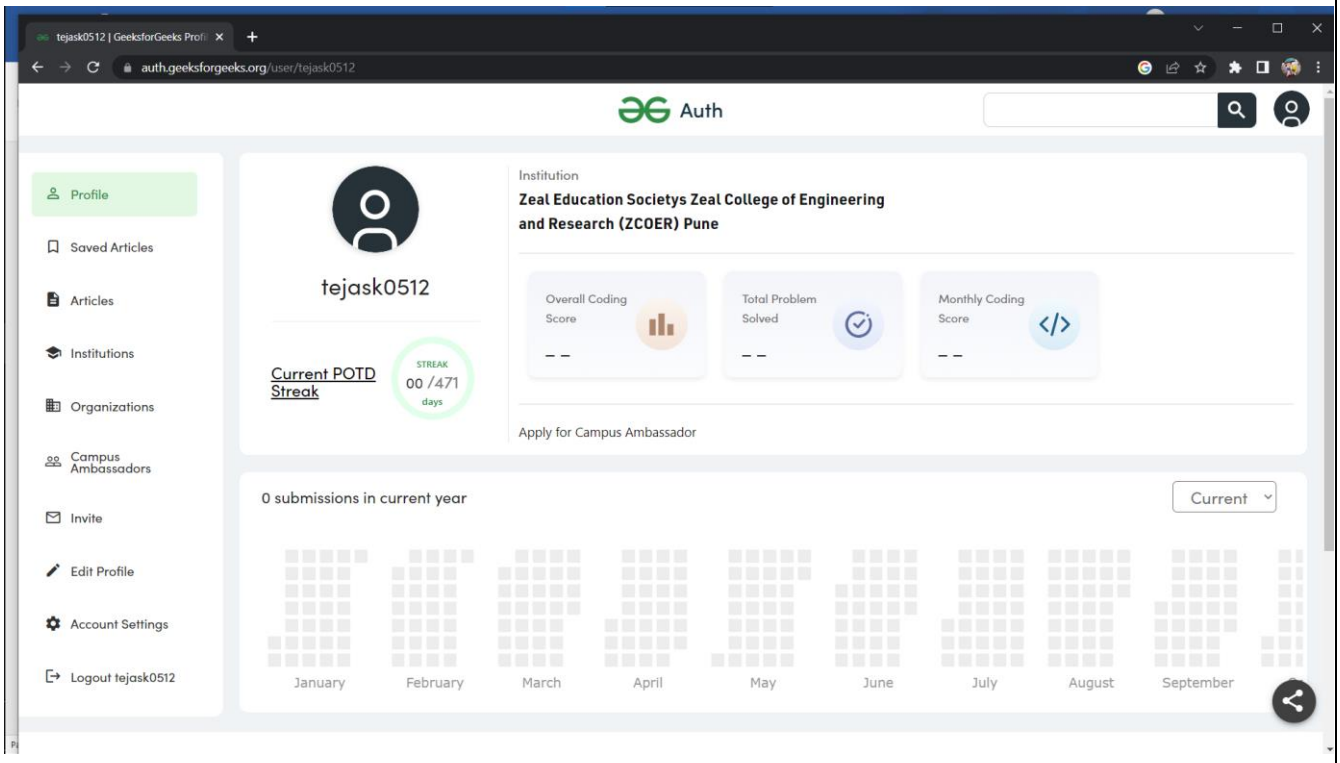

Google
Sign in
Use your Google Account
Email or phone
tejasssssssk
Couldn't find your Google Account
Forgot email?
Not your computer? Use Guest mode to sign in privately.
Learn more
Create account
Next

Assignment No 3

Title: Write test cases in excel sheet for social media application or website.




Please Login To Continue
Sign In
Sign Up
tejask0512@gmail.com
Remember me
Forgot Password
Sign In
or
Google
Facebook
LinkedIn
GitHub
Why Create an Account?
By creating this account, you agree to our Privacy Policy & Cookie Policy.


auth.geeksforgeeks.org/user/tejask0512
Auth
Profile
Saved Articles
Articles
Institutions
Organizations
Campus Ambassadors
Invite
Edit Profile
Account Settings
Logout tejask0512
tejask0512
Institution
Zeal Education Societys Zeal College of Engineering and Research (ZCOER) Pune
Overall Coding Score
Total Problem Solved
Monthly Coding Score
Current POTD Streak
STREAK
00 /471
days
Apply for Campus Ambassador
0 submissions in current year
Current
January
February
March
April
May
June
July
August
September

Assignment No 4

Title: Create defect report for any application or web application.

Sample bug report 2: Application product Bug report sample
Application testing scenario:
Lets assume in your application you want to create a new user with his/her information, for that you need to logon into the application and navigate to USERS menu > New User, then enter all the details in the User form like, First Name, Last Name, Age, Address, Phone etc. Once you enter all these need to click on SAVE button in order to save the user and you can see a success message saying, "New User has been created successfully".
Now you entered into your application by logging in and navigate to USERS menu > New user, entered all the information and clicked on SAVE button and now the application crashed and you can see one error page on the screen, now you would like to report this BUG.

Now here is how we can report bug for above scenario:
Bug Name: Application crash on clicking the SAVE button while creating a new user.
Bug ID: The BUG Tracking tool will automatically create it once you save this.
Area Path: USERS menu > New Users
Build Number:/Version Number 5.0.1
Severity: HIGH (High/Medium/Low)
Priority: HIGH (High/Medium/Low)
Assigned to: Developer-X
Created By: Your Name
Created On: Date
Reason: Defect
Status: New/Open/Active – Depends on the Tool you are using
Environment: Windows 2003/SQL Server 2005
Description:
Application crash on clicking the SAVE button while creating a new user, hence unable to create a new user in the application.
Steps To Reproduce:
1) Logon into the application
2) Navigate to the Users Menu > New User
3) Filled all the fields
4) Clicked on 'Save' button
5) Seen an error page "ORA1090 Exception: Insert values Error…"
6) See the attached logs for more information
7) And also see the attached screenshot of the error page.

Expected: On clicking SAVE button should be prompted to a success message "New User has been created successfully".

Assignment No 5

Title: Prepare software requirement specification for any project or problem statement.

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to build an online system to manage flights and passengers to ease the flight management. <<Include the purpose as applicable to your project >>

1.2 DOCUMENT CONVENTIONS

This document uses the following conventions. <<Include the conventions as per your application >>

DB Database

DDB Distributed Database

ER Entity Relationship

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for the flight management system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the flight management team and as well as to the passengers.

1.4 PROJECT SCOPE

The purpose of the online flight management system is to ease flight management and to create a convenient and easy-to-use application for passengers, trying to buy airline tickets. The system is based on a relational database with its flight management and reservation functions. We will have a database server supporting hundreds of major cities around the world as well as thousands of flights by various airline companies. Above all, we hope to provide a comfortable user experience along with the best pricing available.

1.5 REFERENCES

https://krazytech.com/projects

Fundamentals of database systems by ramezelmarsi and shamkantb.navathe

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

A distributed airline database system stores the following information.

Flight details:

It includes the originating flight terminal and destination terminal, along with the stops in between, the number of seats booked/available seats between two destinations etc.

Customer description:

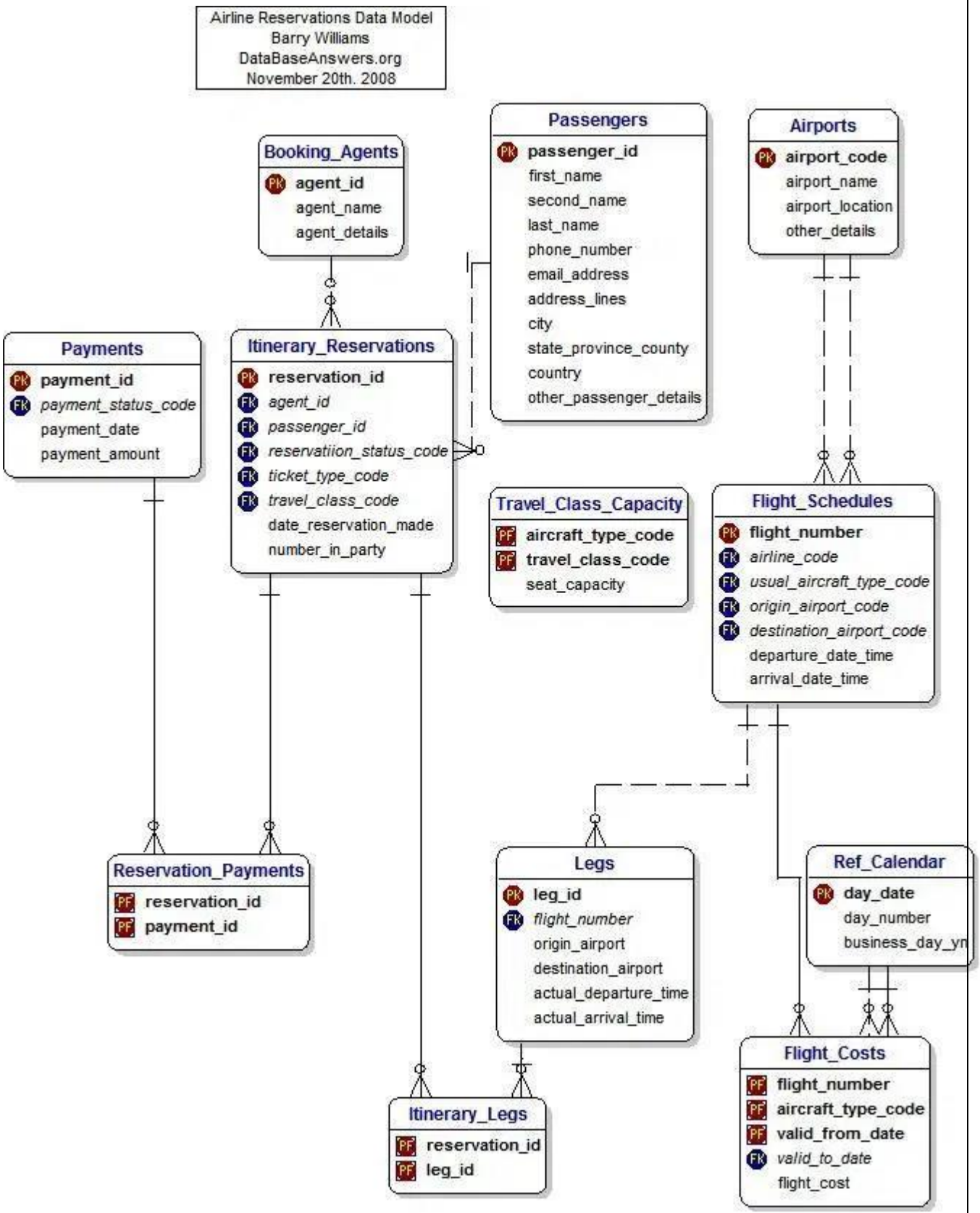
It includes customer code, name, address and phone number. This information may be used for keeping the records of the customer for any emergency or for any other kind of information.

Reservation description:

It includes customer details, code number, flight number, date of booking, date of travel.

2.2 PRODUCT FEATURES

The major features of airline database system as shown in below entity–relationship model (ER model)



layout of airline database system for software requirements specification document

The diagram shows the layout of airline database system – entity–relationship model

2.3 USER CLASS and CHARACTERISTICS

Users of the system should be able to retrieve flight information between two given cities with the given date/time of travel from the database. A route from city A to city B is a sequence of connecting flights from A to B such that: a) there are at most two connecting stops, excluding the starting city and destination city of the trip, b) the connecting time is between one to two hours. The system will support two types of user privileges, Customer, and Employee. Customers will have access to customer functions, and the employees will have access to both customer and flight management functions. The customer should be able to do the following functions:

Make a new reservation

- One-way
- Round-Trip
- Multi-city
- Flexible Date/time
- Confirmation

Cancel an existing reservation

View his itinerary

The Employee should have following management functionalities:

CUSTOMER FUNCTIONS.

- Get all customers who have seats reserved on a given flight.
- Get all flights for a given airport.
- View flight schedule.
- Get all flights whose arrival and departure times are on time/delayed.
- Calculate total sales for a given flight.

ADMINISTRATIVE

- Add/Delete a flight
- Add a new airport
- Update fare for flights.
- Add a new flight leg instance.
- Update departure/arrival times for flight leg instances.

Each flight has a limited number of available seats. There are a number of flights which depart from or arrive at different cities on different dates and time.

2.4 OPERATING ENVIRONMENT

Operating environment for the airline management system is as listed below. <<Include the details as per your application >>

distributed database

client/server system

Operating system: Windows.

database: sql+ database

platform: vb.net/Java/PHP

2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

The global schema, fragmentation schema, and allocation schema.

SQL commands for above queries/applications

How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.

Implement the database at least using a centralized database management system.

2.6 ASSUMPTION DEPENDENCIES

Let us assume that this is a distributed airline management system and it is used in the following application:

A request for booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.

Calculation of high fliers (most frequent fliers) and calculating appropriate reward points for these fliers.

Assuming both the transactions are single transactions, we have designed a distributed database that is geographically dispersed at four cities Delhi, Mumbai, Chennai, and Kolkatta as shown in fig. below.

3. SYSTEM FEATURES

DESCRIPTION and PRIORITY

The airline reservation system maintains information on flights, classes of seats, personal preferences, prices, and bookings. Of course, this project has a high priority because it is very difficult to travel across countries without prior reservations.

STIMULUS/RESPONSE SEQUENCES

Search for Airline Flights for two Travel cities

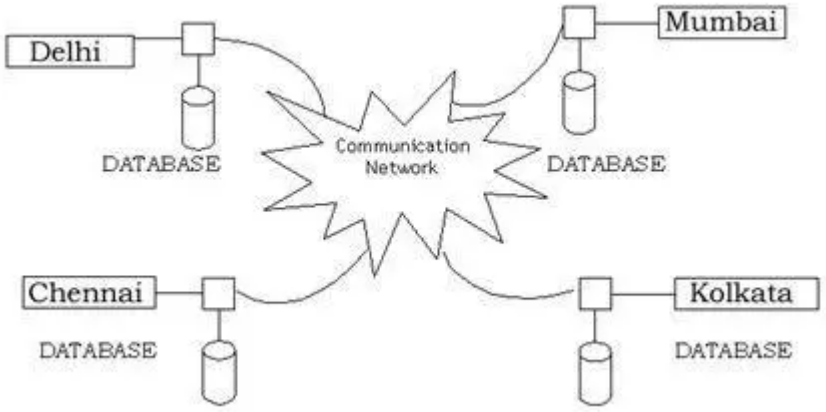
Displays a detailed list of available flights and make a “Reservation” or Book a ticket on a particular flight.

Cancel an existing Reservation.

FUNCTIONAL REQUIREMENTS

Other system features include:

DISTRIBUTED DATABASE:



Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as shown in below figure.

Distributed database located in four different cities

Distributed database located in four different cities

CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

Some sites are client sites and others are server sites.

All the data resides at the server sites.

All applications execute at the client sites.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

Front-end software: Vb.net version

Back-end software: SQL+

4.2 HARDWARE INTERFACES

Windows.

A browser which supports CGI, HTML &Javascript.

4.3 SOFTWARE INTERFACES

Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the flight records, passengers records we have chosen SQL+ database.
VB.Net	To implement the project we have chosen Vb.Net language for its more interactive support.

Following are the software used for the flight management online application. <<Include the software details as per your project >>

4.4 COMMUNICATION INTERFACES

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

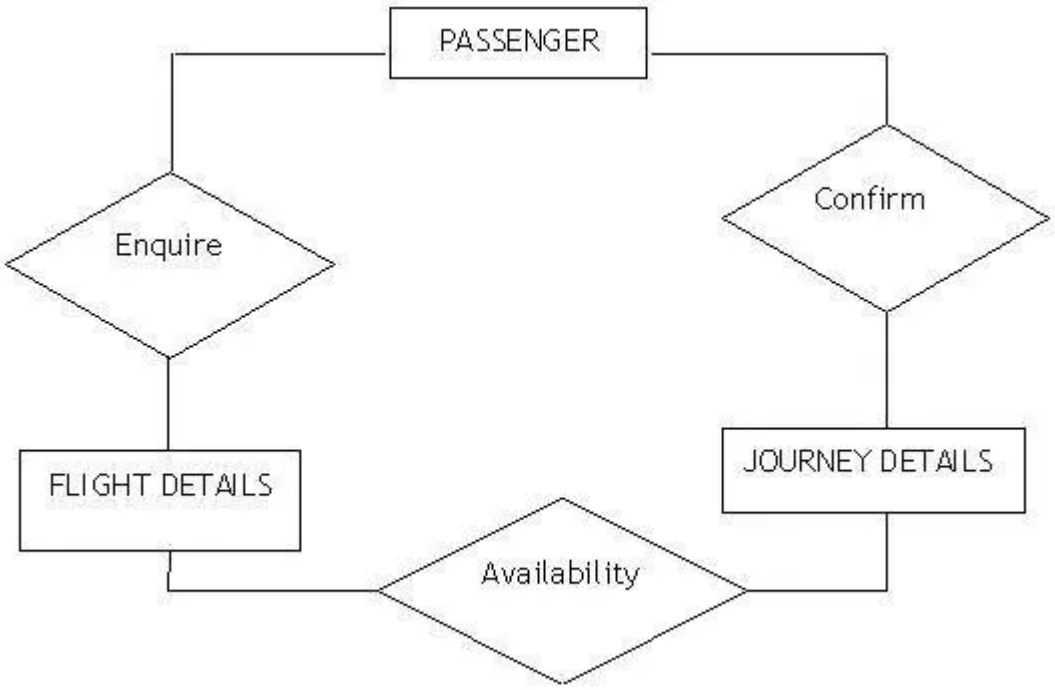
5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of airline database are as listed below.

A) E-R DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.



ENTITIES: Which specify distinct real-world items in an application.

PROPERTIES/ATTRIBUTES: Which specify properties of an entity and relationships.

RELATIONSHIPS: Which connect entities and represent meaningful dependencies between them.

The diagram shows the ER diagram of airline database

the diagram shows the ER diagram of airline database

B) NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4 SOFTWARE QUALITY ATTRIBUTES

AVAILABILITY: The flight should be available on the specified date and specified time as many customers are doing advance reservations.

CORRECTNESS: The flight should reach start from correct start terminal and should reach the correct destination.

MAINTAINABILITY: The administrators and flight in chargers should maintain correct schedules of flights.

USABILITY: The flight schedules should satisfy a maximum number of customers needs.