Project Implementation Report

AUTOMATIC LUGGAGE SUGGESTOR

- **Aditya Chauhan**

School of Information Studies
SYRACUSE UNIVERSITY

Table of Contents

Project Summary....3
Tables and Attributes....4
Business Rules and Assumptions....6
Relational Data Model....7
Database Infrastructure....7
SQL Scripts for Creating and Inserting Sample Data....7
Major Data Questions....12
Relationship Diagram....21
ER Diagram – SQL SERVER....21
Forms and Reports....21
Triggers....30

Luggage Suggestor: Background

There has been a substantial increase in the percentage of people using Online Travel companies to schedule their trip and book their tickets. About 41 % of the tourists prefer to book online tickets through Online Travel Agencies (OTA) like booking.com or Expedia. Also, 50% of people aged between 35 and 44 prefer to reserve through OTAs.

The report found that about 180 million people visited online travel sites in a month. There is a 27 percent increase from the year-earlier period. Online travel agencies are very popular led by Expedia, Priceline, Orbitz and Kayak. In April, 36 million U.S. visitors checked Expedia, the higher than the largest number visiting any individual airline site (Southwest at 16 million) or hotel site (Hilton with 10 million).

The first thing that sparks our mind with the thought of traveling is to research about the place, the temperature at the place and the need to create a checklist of the things that we need to carry. However, in the hustle and bustle, there are often some important items that slip out of our mind. Packing for your trip is too important to put off until the day before you leave because if you're rushing around, you're more likely to forget things that will cost a lot of money to replace during your trip. It's almost inevitable that people forget to pack something when packing for an oncoming trip.

Designed Solution:

- *The user will log in via a Graphical User Interface (GUI)*
- *There will be two different login pages, one for the administrator and one for the traveler*
- *Once the traveler is logged in, the traveler can view the details of the cities and also get a list of luggage they can carry while going to a particular place on the basis of traveling season*
- *The administrator will be able update the location and product details. The administrator have the access to add a user to the database and make him/her an administrator or a traveler. Average temperature of the locations will be pre-added to the database. The temperature will be entered season wise.*
- *The administrator have the access to check the report of inactive users of the application. In an organization this can be of an important use as the organization can target those customers and convince them to use the application to increase the market share.*
- *The administrator have the access to check the report of the most preferred place and the most preferred season. This can be used in an organization to offer lucrative deals to the users.*

TABLES AND ATTRIBUTES:

UserRoles	• **Contains the information of the User Roles (admin and Traveler) and assign a code accordingly.** • **New users roles can be added easily in future**
UserType UserDesc TotalCount	**PRIMARY KEY** : UserType is the primary key of the table having a unique code for each type of user Description of each User Type Count of number of admins and travelers (Based on Trigger. It was added later during trigger creation)
UsersProject	• **Contain information of the users**
UserID FirstName LastName ContactNum Gender emailID Passwrd UserTypeID	**PRIMARY KEY**: Unique ID for each user Primary Contact Number of the User Gender CANDIDATE KEY: Unique Email ID for each User Password **FOREIGN KEY**: User Type of the User. Associated with UserType of the UserRoles table

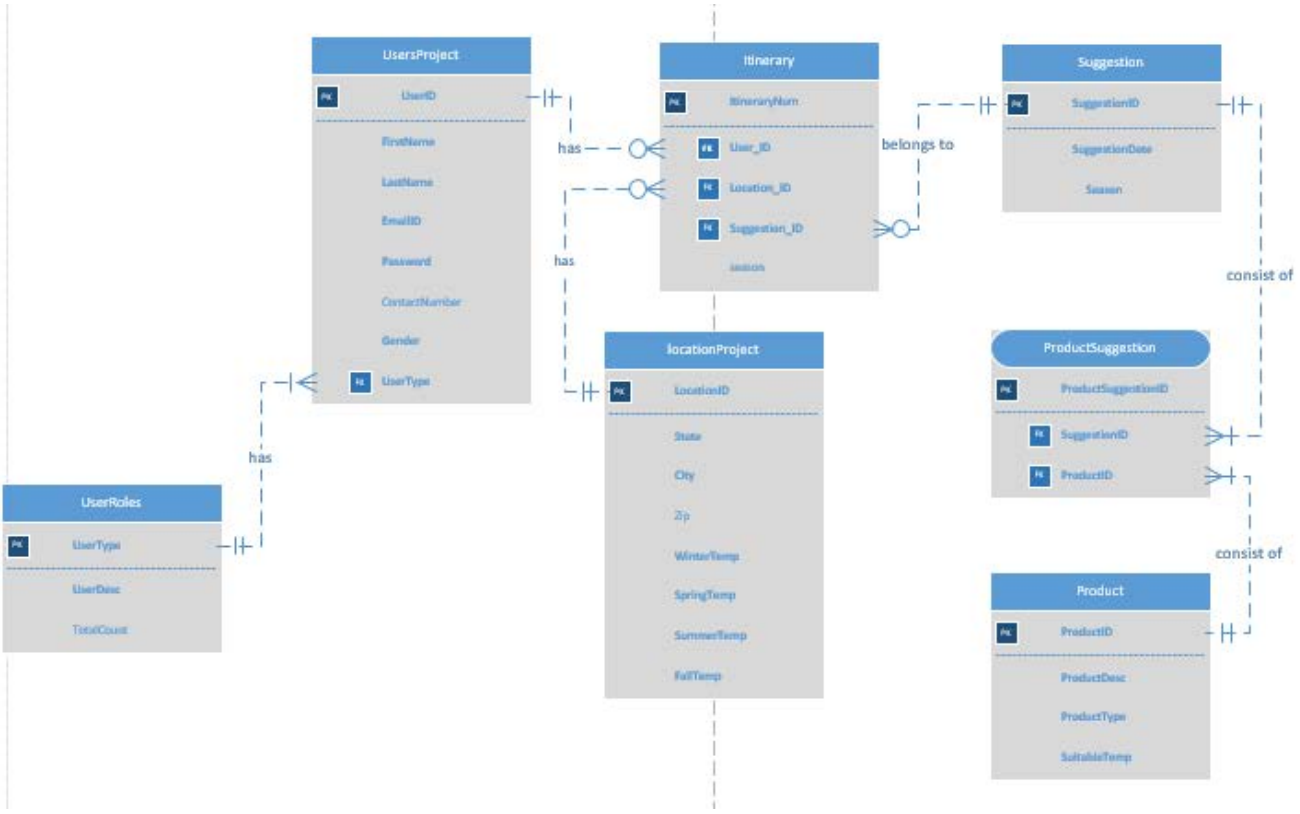
locationProject	• **Contains Location details of theplaces**
LocationID City StateName Zip WinterTemp SpringTemp SummerTemp FallTemp	**PRIMARY KEY**: Each location has a unique ID City Name State Name Zip Code Winter Temperature Spring Temperature Summer Temperature Fall Temperature
ProductTable	• **Contains the list of items that one can use with respect to temperature**
ProductID Productdesc ProductType SuitableTemp	**PRIMARY KEY**: Unique ID of the Apparel that can be wore under a particular minimum temperature Item Description Business or Casual Suitable temperature (in Fahrenheit) at which the apparel can be put on.
Suggestion	• **Holds records of the location and the products that can be carried to the location**
SuggestionID SuggestionDate Season	**PRIMARY KEY**: Unique Suggestions ID of the place with respect to season Date when the suggestion was created Season for which the Suggestion was created (Winter / Spring / Summer / Fall)
ProductSuggestion	• **Mapper Table of the Product and Suggestion Table as they had many to many relationship**
ProductSuggestionID SuggestionID ProductID	**PRIMARY KEY** : Unique ID of the table to hold record **FOREIGN KEY** : Associated with SuggestionID of the Suggestion Table **FOREIGN KEY** : Associated with ProductID of the ProductTable Table

Itinerary	• **This will contain the itinerary of the user**
ItineraryNum	**PRIMARY KEY** : Unique ID of the itinerary created
UserID	**FOREIGN KEY** : Associated with UserID from UsersProject Table
LocationID	**FOREIGN KEY** : Associated with LocationID from locationProject Table
SuggestionID	**FOREIGN KEY**: Item Suggestions based on the details. Associated with SuggestionID from the Suggestion Table
Season	Season in which the user is travelling (1- Winter , 2- Spring, 3- Summer, 4- Fall)

BUSINESS RULES AND ASSUMPTIONS:

- A person can only create one account with one ID and the queries will be answered for only his travel purpose.
- *The administrator can also act as a data analyst. These two are not different User Roles*. An administrator can access all the tables and work as an analyst as well. Administrator can understand the data trends and places of interest among the user.
- The temperature of a place are recorded with respect to seasons i.e Summer, Fall, Spring and Winter
- A user cannot have two roles. A user can only be a Traveler or an Administrator
- Administrator will have access to all the accounts and can add/modify/delete the account.
- Initially the location database will have data of only limited and major cities of United States of America.
- A suggestion may or may not be a part of any Itinerary. More than one itinerary can have same suggestion
- A traveler may or may not visit different cities, and a Location may or may not be visited by one or more travelers
- The ProductType can only be Business or Casual. If the traveler is going for the business trip, suggestions will be given according to business attire.
- All the temperatures mentioned will be in Fahrenheit.
- There will be atleast one Administrator and one Traveler in the UsersProject Table

RELATIONAL DATA MODEL:



Database Infrastructure:

The project is based on the Client – Server architecture i.e each computer or process on the network is either a *client* or a *server*. SQL server is used as the database engine and MS Access is used front end interface tool. Data is inserted, deleted, updated and queried from the SQL server database with the help of forms on Access. Insightful data stored on SQL database can also be viewed with the help of reports generated on MS Access.

SQL SCRIPTS FOR CREATING AND INSERTING SAMPLE DATA:

CREATE: UserRoles

```
CREATE TABLE UserRoles (
userType VARCHAR(10) NOT NULL,
userDesc VARCHAR(10) NOT NULL,
TotalCount INT

CONSTRAINT pk_UserRoles PRIMARY KEY(userType)
);
```

CREATE: UsersProject

```
CREATE TABLE UsersProject (
UserID INT IDENTITY(1,1) NOT NULL,
FirstName VARCHAR(10) NOT NULL,
LastName VARCHAR(10) NOT NULL,
contactNum CHAR(10),
Gender VARCHAR(6) NOT NULL CHECK (Gender IN ('Male', 'Female')),
emailID VARCHAR(20) NOT NULL,
passwrd VARCHAR(20) NOT NULL,
UserType VARCHAR(10) NOT NULL

CONSTRAINT pk_Users  PRIMARY KEY(UserID),
CONSTRAINT fk_Users FOREIGN KEY(UserType) REFERENCES UserRoles(UserType)
);
```

:

CREATE : locationProject

```
CREATE TABLE locationProject (

LocationID INT IDENTITY(1001,1) ,
City VARCHAR(20) NOT NULL,
StateName VARCHAR(20) NOT NULL,
zip INT ,
WinterTemp INT NOT NULL,
SpringTemp INT NOT NULL,
SummerTemp INT NOT NULL,
FallTemp INT NOT NULL,

CONSTRAINT pk_location PRIMARY KEY(LocationID)

)
```

CREATE: Suggestion

```
CREATE TABLE Suggestion (

SuggestionID INT IDENTITY(101,1),
SuggestionDate DATE NOT NULL DEFAULT getdate(),
Season VARCHAR(10) NOT NULL

CONSTRAINT pk_Suggestion PRIMARY KEY (SuggestionID)

)
```

CREATE: ProductTable

```
CREATE TABLE ProductTable (

ProductID INT IDENTITY(200,1),
ProductDesc VARCHAR(30) NOT NULL,
ProductType VARCHAR(10) NOT NULL CHECK (ProductType IN ('Business', 'Casual')),
SuitableTemp INT NOT NULL,

CONSTRAINT pk_Product PRIMARY KEY (ProductID)
)
```

CREATE: productSuggestion

```
CREATE TABLE productSuggestion (

ProductSuggestionID INT IDENTITY(300,1) NOT NULL,
SuggestionID INT NOT NULL,
ProductID INT NOT NULL,

CONSTRAINT pk_ProductSuggestion PRIMARY KEY(ProductSuggestionID),
CONSTRAINT fk_ProductSuggestion FOREIGN KEY (SuggestionID) REFERENCES
Suggestion(SuggestionID),
CONSTRAINT fk_ProductSuggestion2 FOREIGN KEY ( ProductID) REFERENCES ProductTable(ProductID)
)
```

CREATE: Itinerary

```
CREATE TABLE Itinerary (

ItineraryNum INT IDENTITY(5000,1) NOT NULL,
UserID INT NOT NULL,
LocationID INT NOT NULL,
SuggestionID INT NOT NULL,
Season VARCHAR(10)

CONSTRAINT pk_Itinerary PRIMARY KEY(ItineraryNum),
CONSTRAINT fk_Itinerary FOREIGN KEY (UserID) REFERENCES UsersProject(UserID),
CONSTRAINT fk_Itinerary2 FOREIGN KEY (LocationID) REFERENCES locationProject(LocationID),
CONSTRAINT fk_Itinerary3 FOREIGN KEY (SuggestionID) REFERENCES Suggestion(SuggestionID)
)
```

INSERT DATA: UserRoles

Initially the table was created with only 2 attributes, later a third column was added to determine the count of Administrator and Traveler in the database which is updated automatically through trigger

```
INSERT INTO UserRoles VALUES ('R1', 'Admin');
INSERT INTO UserRoles VALUES ('R2', 'Traveler');
```

```
SELECT * FROM UserRoles;
```

	userType	userDesc	TotalCount
1	R1	Admin	5
2	R2	Traveler	22

INSERT DATA: UsersProject

```
INSERT INTO UsersProject VALUES
('Aditya','Chauhan',3157514953,'Male','adi1909@live.com','Welcome123','R1')
INSERT INTO UsersProject VALUES
('Aditya','Chauhan',8130681577,'Male','achauh01@xyz.com','Hello123','R2')
```

Rest of the values in the table were entered through User Addition Form

Below is the snippet of some of the values from the table

```
SELECT * FROM UsersProject;
```

	UserID	FirstName	LastName	contactNum	Gender	emailID	passwrd	UserTy
1	1	Aditya	Chauhan	3157514953	Male	adi1909@live.com	Welcome123	R1
2	2	Aditya	Chauhan	8130681577	Male	achauh01@xyz.com	Hello123	R2
3	3	Nitin	Khanna	8393930010	Male	nkhanna@gmail.com	Nitin123	R2
4	4	Priyam	Shukla	3991901002	Female	pshukla@gmail.com	Priyam123	R2
5	5	Sachin	Tendulkar	7271829378	Male	stendulkar@gmail.com	Sachin123	R2
6	6	Issac	Newton	3783010022	Male	inewton@gmail.com	Issac123	R1
7	7	Christiano	Ronaldo	3974831038	Male	cronaldo@gmail.com	Ronaldo123	R2
8	9	Ranbeer	Kapoor	3738129303	Male	rkapoor@gmail.com	Ranbeer123	R2
9	13	Karan	Johar	3738310011	Male	kjohar@gmail.com	Karan123	R1
10	14	Yun	Huang	2818203003	Female	yhuang@syr.edu	Welcome123	R1
11	15	Anmol	Handa	3157514953	Male	anmolhanda@gmail....	Anmol123	R2
12	16	Aditi	Chawla	8323302001	Female	adchawla@syr.edu	Aditi123	R2
13	17	Arjun	Suri	3157514958	Male	arsuri@gmail.com	Arjun123	R2
14	18	Pooja	Shah	2389320201	Female	pshah@gmail.com	Pooja123	R1
15	19	Niti	Saluja	3873932010	Female	nsaluja@gmail.com	Niti123	R2
16	20	John	Barnfield	3174895009	Male	jbarnfield@gmail.com	John123	R2
17	21	Katrina	Kaif	8849404030	Female	kkaif@gmail.com	Katrina123	R2
18	22	Soumya	Samal	7383903039	Male	ssamal@gmail.com	Soumya123	R2
19	23	Mandeep	Rana	3874940400	Male	mrana@gmail.com	Mandeep123	R2
20	24	Gauri	Gajaral...	2346657873	Female	gauri@gmail.com	Gauri123	R2
21	26	Rajesh	A	8494040480	Male	rajesh@gmail.com	Rajesh123	R2
22	27	Apoorva	Angre	4849404048	Female	aangre@gmail.com	Apoorva123	R2
23	28	Nidhi	Rakheja	4743849404	Female	nrakheja@gmail.com	Nidhi123	R2

INSERT DATA: locationProject

```
INSERT INTO locationProject VALUES ('Albany','New York',12201,22,46,71,49)
INSERT INTO locationProject VALUES ('Atlanta','Georgia',30301,42,61,80,62)
```

Rest of the values in the table were entered through Location Form

Below is the snippet of some of the values from the table

	LocationID	City	StateName	zip	WinterTemp	SpringTemp	SummerTemp	FallTemp
1	1001	Albany	New York	12201	22	46	71	49
2	1002	Atlanta	Georgia	30301	42	61	80	62
3	1003	Atlantic City	New jersey	8201	32	50	75	55
4	1004	Austin	Texas	73301	50	68	84	70
5	1005	Baltimore	Maryland	21201	32	53	76	55
6	1006	Birmingham	Alabama	35005	42	61	80	70
7	1007	Boston	Massachusetts	1841	29	48	73	54
8	1008	Buffalo	New York	14201	24	45	70	50
9	1009	Charlotte	North Carolina	28105	41	60	80	61
10	1010	Syracuse	New York	13210	22	48	80	50

INSERT DATA: ProductTable

```
INSERT INTO ProductTable VALUES('Shorts','Casual',70)
INSERT INTO ProductTable VALUES('Short-Sleeve Jersey','Casual',70)
```

Rest of the values in the table were entered through **Product Form**

Below is the snippet of some of the values from the table

	ProductID	ProductDesc	ProductType	SuitableTemp
31	230	Lined Skull Cap	Casual	30
32	231	Winter Bib Tights	Casual	25
33	232	Mittens	Casual	25
34	233	Lobster Claw Gloves	Casual	25
35	234	Balaclava	Casual	25
36	235	Plastic Bag	Casual	25
37	236	Tshirt	Casual	70

INSERT DATA: Suggestion

```
INSERT INTO Suggestion VALUES('09/12/2017','Winter');
INSERT INTO Suggestion VALUES('08/23/2017','Spring');
INSERT INTO Suggestion VALUES('07/25/2017','Summer');
INSERT INTO Suggestion VALUES('09/18/2017','Fall');
```

	SuggestionID	SuggestionDate	Season
1	101	2017-09-12	Winter
2	102	2017-08-23	Spring
3	103	2017-07-25	Summer
4	104	2017-09-18	Fall

INSERT DATA: ProductSuggestion

```
INSERT INTO ProductSuggestion VALUES(103,200)
INSERT INTO ProductSuggestion VALUES(103,201)
INSERT INTO ProductSuggestion VALUES(101,205)
```

Rest of the values in the table were entered through **ProductSuggestion Form**

Below is the snippet of some of the values from the table

	ProductSuggestionID	SuggestionID	ProductID
1	300	103	200
2	301	103	201
3	302	103	203
4	303	101	205
5	304	101	206
6	305	101	207
7	306	101	210

INSERT DATA: Itinerary

```
INSERT INTO itinerary VALUES(2,1004,101,1)
INSERT INTO itinerary VALUES(9,1003,101,1)
```

Rest of the values in the table were created through **Itinerary Form in the Traveler's Page.**

Below is the snippet of some of the values from the table

	ItineraryNum	UserID	LocationID	SuggestionID	season
1	5022	2	1004	101	1
2	5023	4	1004	101	1
3	5024	9	1003	101	1
4	5025	19	1007	101	1
5	5026	4	1003	102	2
6	5027	32	1005	104	4
7	5028	24	1006	103	3
8	5029	31	1004	101	1
9	5030	23	1009	101	1
10	5031	3	1006	104	4

MAJOR DATA QUESTIONS:

The users using the system are of three types:

1) User (Traveler)

2) Administrator

Following list shows segregation of what data questions arises and its detailed explanation.

What is the source of the temperature Database?

The recorded average seasonly temperatures values will be taken from http://weatherbase.com . Weatherbase also offers a comprehensive weather resource center, including a weather glossary and our Vacation Finder, a unique planning technology that finds places to go based on the average weather in cities worldwide.

Why Traveler queries the database?

Traveler queries the database to get the list of the items that he can pack while going for a trip. To get the desired results, a traveler will be asked to enter the Name of the place, and season of travel, on the basis of which the results will be shown. The 'locationProject' table will determine the name of the place having details of the average seasonal temperatures. The list of items in the database are entered on the basis of the suitable temperature it can be worn at.

Also, there are additional attributes such as Product Description (ProductDesc) , Product Type(ProductType) such as Business or casuals, and the suitable temp (SuitableTemp) at which the attire can be worn.

'locationProject' Table holds the record of the database of the cities along with the state and ZIP Codes. The table also holds information of the seasonal average temperatures. Location table will be used to fetch details of the location which is visited by the traveler and pull in Itinerary table.

Also, the administrator can determine the change in weather trends at a place. When implemented on a larger scale, the administrator can learn about the most preferred place and get in touch with the online travel industry to offer lucrative deals and attract the user.

- Get the Itinerary when travelling in Winter Season

The Report of this business question is in the Forms and Reports section of the document.

```
SELECT dbo_ProductTable.ProductDesc
FROM dbo_ProductTable INNER JOIN dbo_ProductSuggestion ON dbo_ProductTable.ProductID=dbo_ProductSuggestion.ProductID
WHERE dbo_ProductSuggestion.SuggestionID = 101;
```

Leg Warmer
Heavy Long Sleeve Jersey
Short Sleeve Wicking Undershir
Thin Full fingered gloves
Headband covering ears
Wool Socks
Shoe Covers
Heavy Weight tights
Heavy Cycling Jacket
Heavy Weight Gloves
Charcoel Toe Warmers
Lined Skull Cap
Lobster Claw Gloves
Balaclava
Plastic Bag
Tshirt
Shirt
Jeans

```
SELECT ProductTable.ProductDesc
FROM ProductTable INNER JOIN ProductSuggestion ON ProductTable.ProductID=ProductSuggestion.ProductID
WHERE ProductSuggestion.SuggestionID = 101;
```

	ProductDesc
1	Leg Warmer
2	Heavy Long Sleeve Jersey
3	Short Sleeve Wicking Undershir
4	Thin Full fingered gloves
5	Headband covering ears
6	Wool Socks
7	Shoe Covers
8	Heavy Weight tights
9	Heavy Cycling Jacket
10	Heavy Weight Gloves
11	Charcoel Toe Warmers
12	Lined Skull Cap
13	Lobster Claw Gloves
14	Balaclava
15	Plastic Bag
16	Tshirt
17	Shirt

- Get the Itinerary when travelling in Summer Season

The Report of this business question is in the Forms and Reports section of the document.

```
SELECT dbo_ProductTable.ProductDesc
FROM dbo_ProductTable INNER JOIN dbo_ProductSuggestion ON dbo_ProductTable.ProductID=dbo_ProductSuggestion.ProductID
WHERE dbo_ProductSuggestion.SuggestionID = 103;
```

Shorts
Short-Sleeve Jersey
Long Sleeve Jersey
Tshirt
Shirt
Jeans

```
SELECT ProductTable.ProductDesc
FROM ProductTable INNER JOIN ProductSuggestion ON ProductTable.ProductID=ProductSuggestion.ProductID
WHERE ProductSuggestion.SuggestionID = 103;
```

	ProductDesc
1	Shorts
2	Short-Sleeve Jersey
3	Long Sleeve Jersey
4	Tshirt
5	Shirt
6	Jeans

- Get the Itinerary when travelling in Spring Season

The Report of this business question is in the Forms and Reports section of the document.

```
SELECT dbo_ProductTable.ProductDesc
FROM dbo_ProductTable INNER JOIN dbo_ProductSuggestion ON dbo_ProductTable.ProductID=dbo_ProductSuggestion.ProductID
WHERE dbo_ProductSuggestion.SuggestionID = 102;
```

ProductDesc
Shorts
Long Sleeve Jersey
Long Sleeve Thin Undershirt
Thin Full fingered gloves
Medium Weight Gloves
Tshirt
Shirt
Jeans
Raincoat
Rainboot

```
SELECT ProductTable.ProductDesc
FROM ProductTable INNER JOIN ProductSuggestion ON ProductTable.ProductID=ProductSuggestion.ProductID
WHERE ProductSuggestion.SuggestionID = 102;
```

	ProductDesc
1	Shorts
2	Long Sleeve Jersey
3	Long Sleeve Thin Undershirt
4	Thin Full fingered gloves
5	Medium Weight Gloves
6	Tshirt
7	Shirt
8	Jeans
9	Raincoat
10	Rainboot

- Get the Itinerary when travelling in Fall Season

The Report of this business question is in the Forms and Reports section of the document.

```
SELECT dbo_ProductTable.ProductDesc
FROM dbo_ProductTable INNER JOIN dbo_ProductSuggestion ON dbo_ProductTable.ProductID=dbo_ProductSuggestion.ProductID
WHERE dbo_ProductSuggestion.SuggestionID = 104;
```

ProductDesc
Short-Sleeve Jersey
Long Sleeve Thin Undershirt
Tshirt
Shirt
Jeans
Raincoat
Rainboot
Sweatshirts
Light Jacket

```
SELECT ProductTable.ProductDesc
FROM ProductTable INNER JOIN ProductSuggestion ON ProductTable.ProductID=ProductSuggestion.ProductID
WHERE ProductSuggestion.SuggestionID = 104;
```

	ProductDesc
1	Short-Sleeve Jersey
2	Long Sleeve Thin Undershirt
3	Tshirt
4	Shirt
5	Jeans
6	Raincoat
7	Rainboot
8	Sweatshirts
9	Light Jacket

Show the list of cities a user may travel

```
SELECT dbo_locationProject.City, dbo_locationProject.StateName AS ['State Name'], dbo_locationProject.zip, dbo_locationProject.WinterTemp AS [' Winter Temp'],
dbo_locationProject.SpringTemp AS [' Spring Temp'], dbo_locationProject.SummerTemp AS [' Summer Temp'], dbo_locationProject.FallTemp AS [' Fall Temp']
FROM dbo_locationProject;
```

City	'State Name'	zip	' Winter Tem	' Spring Tem	' Summer Te	' Fall Temp'
Albany	New York	12201	22	46	71	49
Atlanta	Georgia	30301	42	61	80	62
Atlantic City	New jersey	8201	32	50	75	55
Austin	Texas	73301	50	68	84	70
Baltimore	Maryland	21201	32	53	76	55
Birmingham	Alabama	35005	42	61	80	70
Boston	Massachusetts	1841	29	48	73	54
Buffalo	New York	14201	24	45	70	50
Charlotte	North Carolina	28105	41	60	80	61

```
SELECT locationProject.City, locationProject.StateName AS 'State Name', locationProject.zip,
locationProject.WinterTemp AS ' Winter Temp', locationProject.SpringTemp AS ' Spring Temp',
locationProject.SummerTemp AS ' Summer Temp', locationProject.FallTemp AS ' Fall Temp'
FROM locationProject;
```

	City	State Name	zip	Winter Temp	Spring Temp	Summer Temp	Fall Temp
1	Albany	New York	12201	22	46	71	49
2	Atlanta	Georgia	30301	42	61	80	62
3	Atlantic City	New jersey	8201	32	50	75	55
4	Austin	Texas	73301	50	68	84	70
5	Baltimore	Maryland	21201	32	53	76	55
6	Birmingham	Alabama	35005	42	61	80	70
7	Boston	Massachusetts	1841	29	48	73	54
8	Buffalo	New York	14201	24	45	70	50
9	Charlotte	North Carolina	28105	41	60	80	61

Why administrator queries the database?

Administrator queries the database to assign roles to the users. The Administrator will be having access to add/modify/delete the user from the database. Also, database administrator will be able to update the average temperatures and add the entries for the new places.

If the details of any user are wrongly entered, the administrator will be having access to update the role.

The administrator can also create a record of the user under the User Table and assign a role from the UserRole table.

An administrator can also act as data analyst. Administrator will be able to access the Itinerary data and shall determine the trend of the count of people visiting a city in a particular season. With the data in hand, the analyst will be able to analyze the most preferred city amongst the travelers for a trip and their preferable season of travel.

- Which are the most preferred cities among users to travel

The Report of this business question is in the Forms and Reports section of the document.

```
SELECT dbo_locationProject.City, Count(dbo_Itinerary.LocationID) AS CountOfLocationID
FROM dbo_Itinerary INNER JOIN dbo_locationProject ON dbo_itinerary.LocationID=dbo_locationProject.LocationID
GROUP BY dbo_locationProject.City;
```

City	CountOfLocationID
Atlantic City	2
Austin	3
Baltimore	1
Birmingham	2
Boston	1
Charlotte	1

```
SELECT locationProject.City, Count(Itinerary.LocationID) AS 'Count Of LocationID'
FROM Itinerary INNER JOIN locationProject ON Itinerary.LocationID=locationProject.LocationID
GROUP BY locationProject.City;
```

	City	Count Of LocationID
1	Atlantic City	2
2	Austin	3
3	Baltimore	1
4	Birmingham	2
5	Boston	1
6	Charlotte	1

- Which is the most preferred season among users to travel?

The Report of this business question is in the Forms and Reports section of the document.

```
SELECT dbo_Suggestion.Season, Count(dbo_Itinerary.SuggestionID) AS [Count]
FROM dbo_Itinerary INNER JOIN dbo_Suggestion ON dbo_Suggestion.SuggestionID=dbo_Itinerary.SuggestionID
GROUP BY dbo_Suggestion.Season;
```

Season	Count
Fall	2
Spring	1
Summer	1
Winter	6

```
SELECT Suggestion.Season, Count(Itinerary.SuggestionID) AS 'Count'
FROM Itinerary INNER JOIN Suggestion ON Suggestion.SuggestionID=Itinerary.SuggestionID
GROUP BY Suggestion.Season;
```

	Season	Count
1	Fall	2
2	Spring	1
3	Summer	1
4	Winter	6

- Which are the most active users?

The Report of this business question is in the Forms and Reports section of the document.

```
SELECT dbo_UsersProject.FirstName, dbo_UsersProject.LastName, Count(dbo_UsersProject.UserID) AS CountOfUserID
FROM dbo_itinerary INNER JOIN dbo_UsersProject ON dbo_itinerary.UserID = dbo_UsersProject.UserID
GROUP BY dbo_UsersProject.FirstName, dbo_UsersProject.LastName, dbo_UsersProject.UserID
ORDER BY dbo_UsersProject.UserID;
```

FirstName	LastName	CountOfUser
Aditya	Chauhan	1
Nitin	Khanna	1
Priyam	Shukla	2
Ranbeer	Kapoor	1
Niti	Saluja	1
Mandeep	Rana	1
Gauri	Gajaralwar	1
Zhihui	Hong	1
Sagar	Bhojwani	1

```
SELECT UsersProject.FirstName, UsersProject.LastName, Count(UsersProject.UserID) AS 'Count Of UserID'
FROM Itinerary INNER JOIN UsersProject ON Itinerary.UserID = UsersProject.UserID
GROUP BY UsersProject.FirstName, UsersProject.LastName, UsersProject.UserID
ORDER BY UsersProject.UserID;
```

	FirstName	LastName	Count Of UserID
1	Aditya	Chauhan	1
2	Nitin	Khanna	1
3	Priyam	Shukla	2
4	Ranbeer	Kapoor	1
5	Niti	Saluja	1
6	Mandeep	Rana	1
7	Gauri	Gajaralwar	1
8	Zhihui	Hong	1
9	Sagar	Bhojwani	1

- Which are the most inactive users / users who didn't have the itinerary created?

```
SELECT dbo_UsersProject.FirstName, dbo_UsersProject.LastName, dbo_UsersProject.emailID
FROM dbo_UsersProject LEFT JOIN dbo_Itinerary ON dbo_UsersProject.UserID = dbo_Itinerary.UserID
WHERE dbo_Itinerary.UserID IS NULL AND dbo_UsersProject.UserType = "R2";
```

FirstName	LastName	emailID
Sachin	Tendulkar	stendulkar@gmail.com
Christiano	Ronaldo	cronaldo@gmail.com
Anmol	Handa	anmolhanda@gmail.com
Aditi	Chawla	adchawla@syr.edu
Arjun	Suri	arsuri@gmail.com
John	Barnfield	jbarnfield@gmail.com
Katrina	Kaif	kkaif@gmail.com
Soumya	Samal	ssamal@gmail.com
Rajesh	A	rajesh@gmail.com
Apoorva	Angre	aangre@gmail.com
Nidhi	Rakheja	nrakheja@gmail.com
Roshan	Kumar	rkumar@gmail.com
Ananya	B	ananya@gmail.com

```
SELECT UsersProject.FirstName, UsersProject.LastName, UsersProject.emailID
FROM UsersProject LEFT JOIN Itinerary ON UsersProject.UserID = Itinerary.UserID
WHERE Itinerary.UserID IS NULL AND UsersProject.UserType = 'R2';
```

	FirstName	LastName	emailID
1	Sachin	Tendulkar	stendulkar@gmail.com
2	Christiano	Ronaldo	cronaldo@gmail.com
3	Anmol	Handa	anmolhanda@gmail.com
4	Aditi	Chawla	adchawla@syr.edu
5	Arjun	Suri	arsuri@gmail.com
6	John	Barnfield	jbarnfield@gmail.com
7	Katrina	Kaif	kkaif@gmail.com
8	Soumya	Samal	ssamal@gmail.com
9	Rajesh	A	rajesh@gmail.com
10	Apoorva	Angre	aangre@gmail.com
11	Nidhi	Rakheja	nrakheja@gmail.com
12	Roshan	Kumar	rkumar@gmail.com
13	Ananya	B	ananya@gmail.com

- Get the User Statistics of how many administrators and travelers are there in the database?

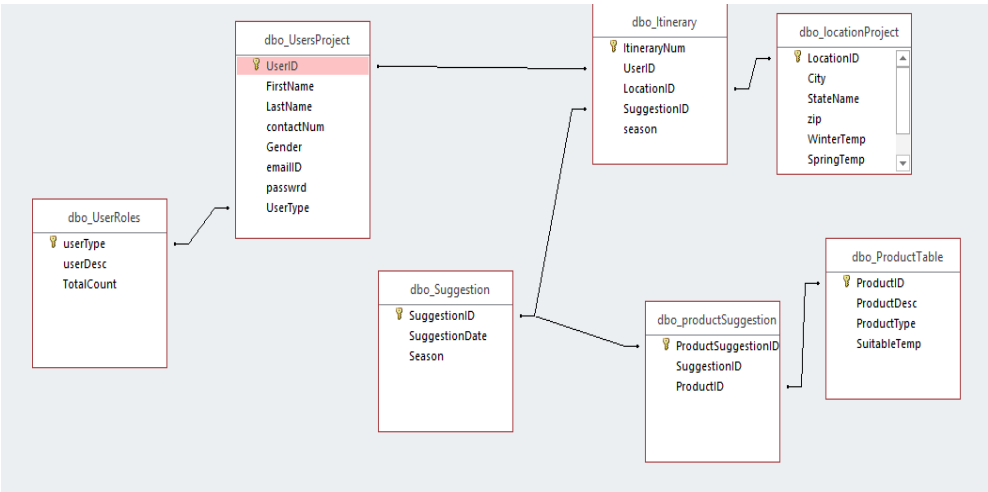
This shows the contents of the UserRoles table which gets updated using the trigger which is discussed in detail in the last section of the report. Below is the screenshot of the UserRoles Table. The Report of this business question is in the Forms and Reports section of the document.

```
SELECT * FROM UserRoles;
```

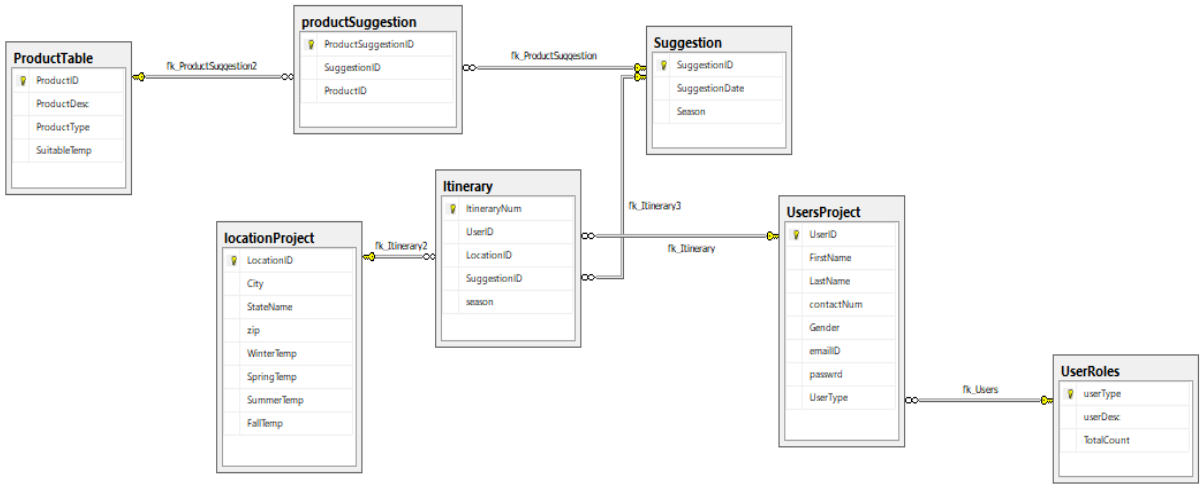
	userType	userDesc	TotalCount
1	R1	Admin	5
2	R2	Traveler	22

RELATIONSHIP DIAGRAM:

The Relationship Diagram is as under:



ER Diagram – SQL Server



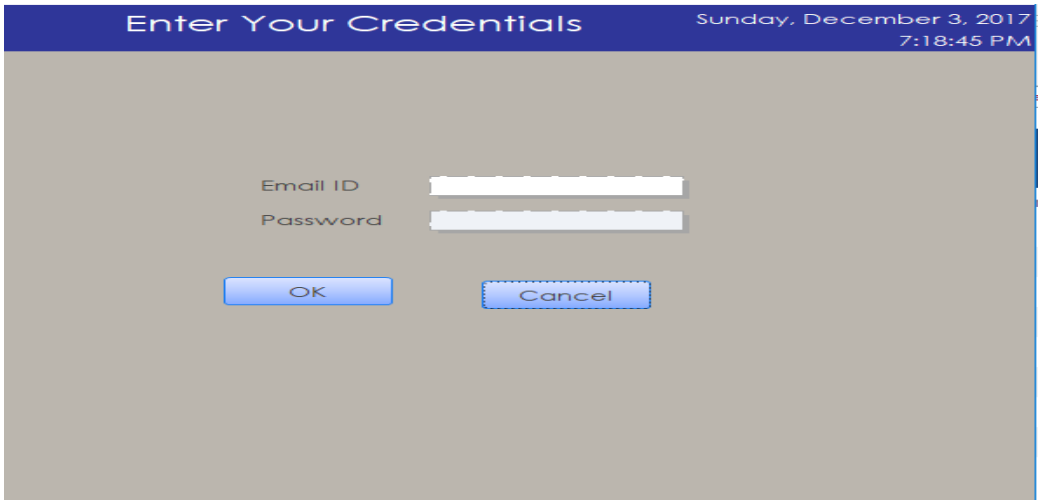
FORMS AND REPORTS:

The Video Caption Database has two type of users:

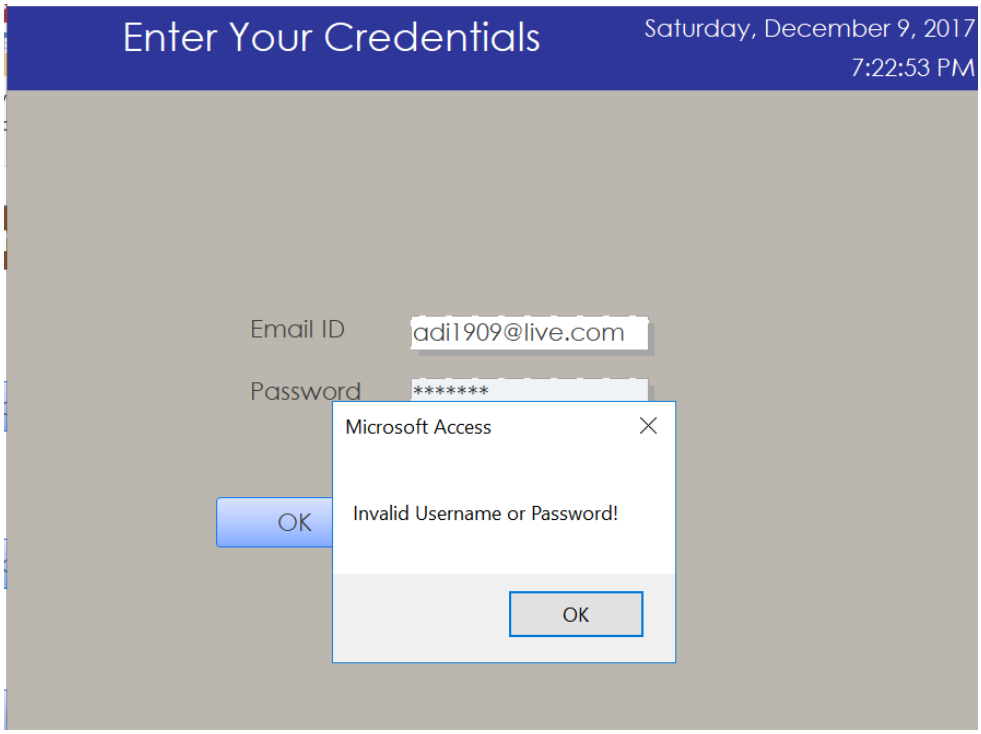
1. The Administrator user: They have the admin rights to the database and can read/write to the Database.

2. The Student User: They have limited access to the database.

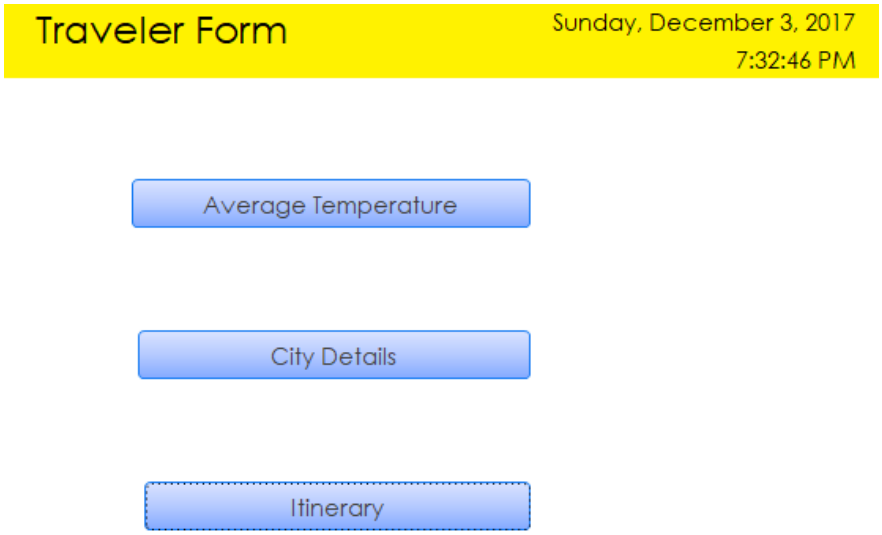
This is the login form, where the user will be prompted to enter his Email ID and password.



On Entering correct username and password, access will be granted to the user and depending on the type of user different home screens will appear. Proper validation has been implemented to ensure the security.



Traveler Page:



Step 1: Click on Average Temperature **to check the annual average temperature of the places**

Average Temperature of Cities

City	Average Temp
Albany	47
Atlanta	61.25
Atlantic City	53
Austin	68
Baltimore	54
Birmingham	63.25
Boston	51
Buffalo	47.25
Charlotte	60.5

Sunday, December 3, 2017 Page 1 of 1

Step 2: Click on City Details **to fetch the city details along with season wise temperatures.**

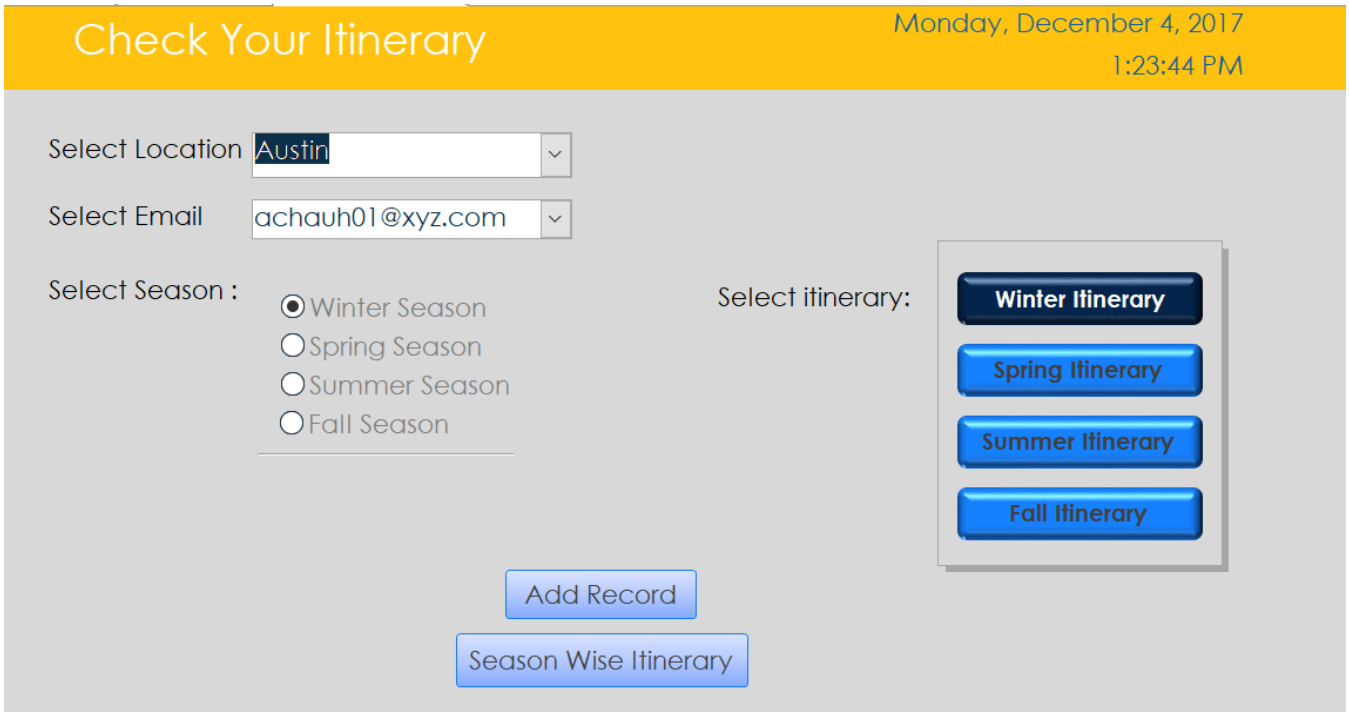
City Details

City	State Name	Zip	Winter Temp	Spring Temp	Summer Temp	Fall Temp
Albany	New York	12201	22	46	71	49
Atlanta	Georgia	30301	42	61	80	62
Atlantic City	New jersey	8201	32	50	75	55
Austin	Texas	73301	50	68	84	70
Baltimore	Maryland	21201	32	53	76	55
Birmingham	Alabama	35005	42	61	80	70
Boston	Massachusetts	1841	29	48	73	54
Buffalo	New York	14201	24	45	70	50
Charlotte	North Carolina	28105	41	60	80	61

Monday, December 4, 2017 Page 1 of 1

Step 3: Search for Itinerary / suggestions. Click on 

to get the list of suggestions:



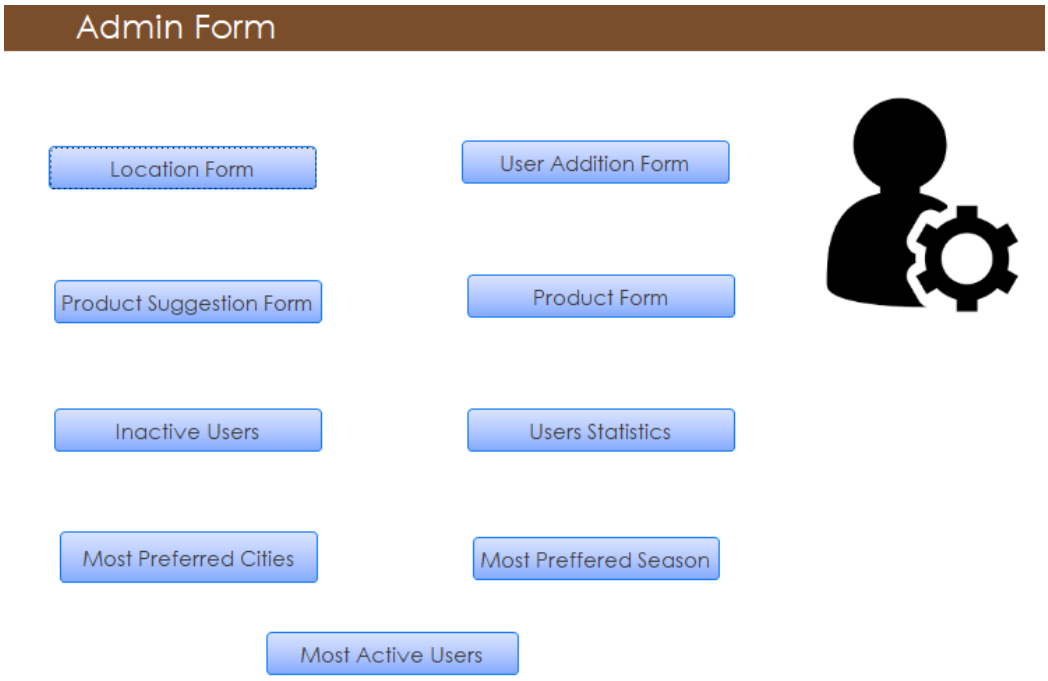
Traveler need to do the following sequence of steps in order to get the suggestions.

1) Selection Location
2) Select Email

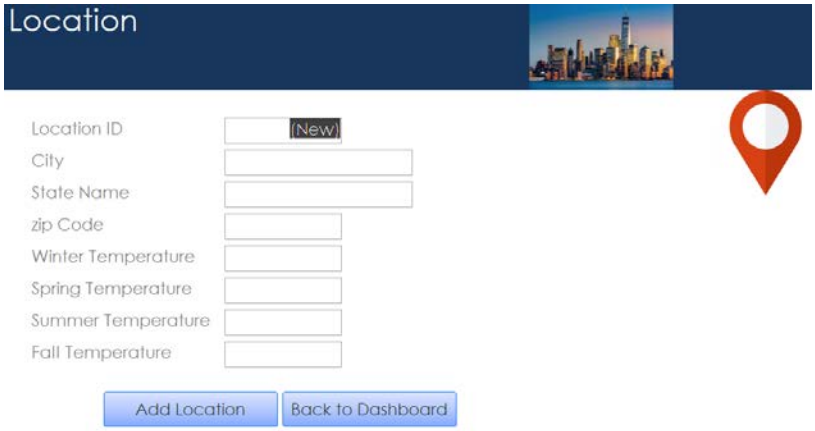
3) Select Season
4) Select Itinerary
5) Click on Add Record : This will add the record to the itinerary table which can later be analyzed by the administrator
6) Click on Season Wise Itinerary: This will take the traveler to the below page.

7) Traveler need to click on the Itinerary he wish to see. For example : Itinerary might be one from the below 4 reports of each season:

Administrator Page:



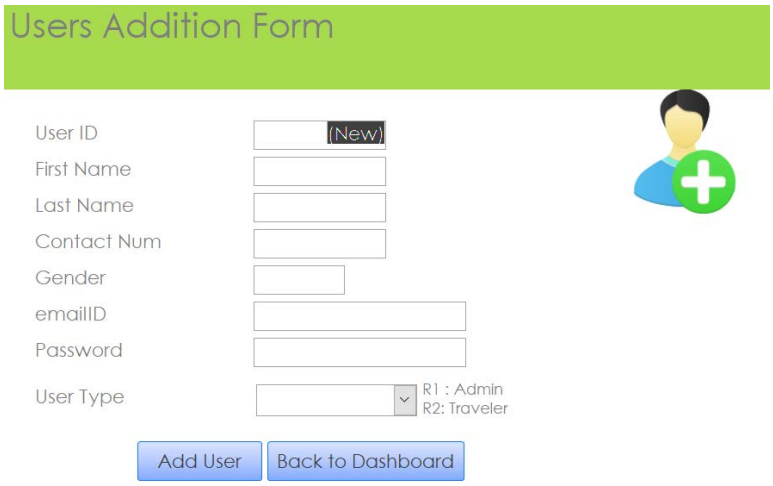
Feature 1: An administrator can create/update/delete location details using Location Form. Once the administrator click on the Location Form, he will get access to the below mentioned page where he/she can do the amendments.



Administrator needs to add the location details and click on "Add Location" to add the record. By Clicking on "Back to Dashboard "Page, administrator will be routed back to Administrator form.

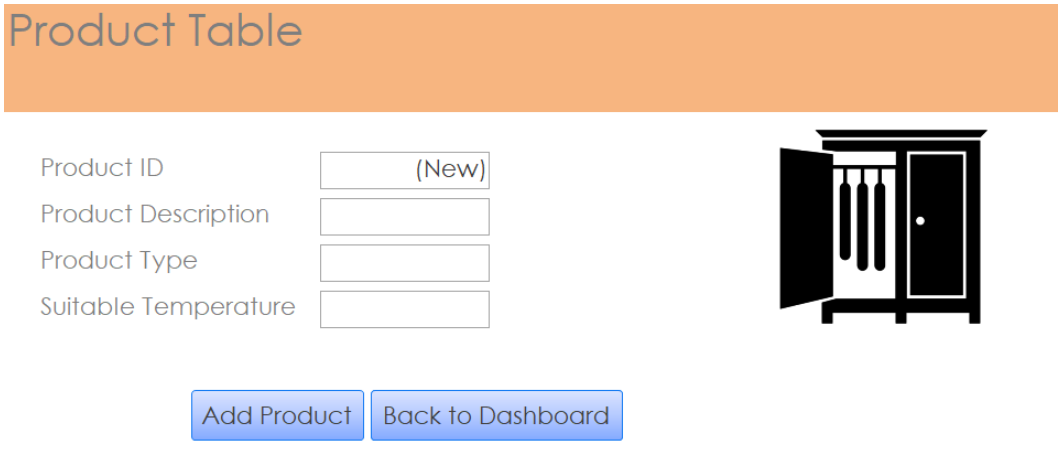
Feature 2: An administrator can create/update/modify Users Information in the database and assign roles to them. R1 is for Admin, whereas, R2 is for Traveler. The password entered will be shown as

"*******" to maintain privacy. To make the amendment administrator need to click on User Addition Form. On clicking this button, administrator will be routed to the below mentioned page.

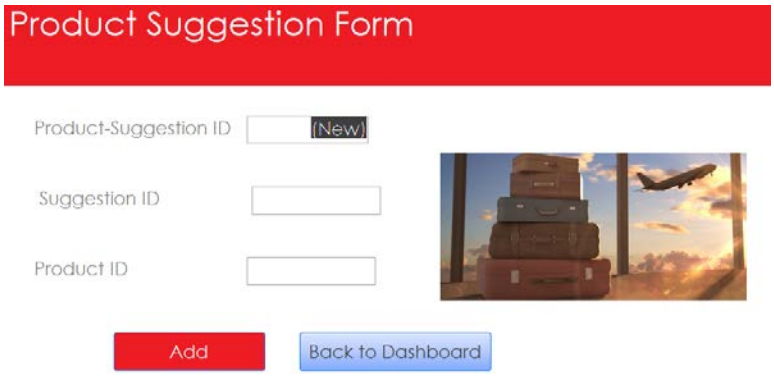


Administrator needs to add the User details and click on "Add User" to add the record. By Clicking on "Back to Dashboard "Page, administrator will be routed back to Administrator form.

Feature 3: An administrator can create /update/modify Product information in the database. To make the amendment, administrator need to click on Product Form. On clicking this button, the administrator will be routed to the below mentioned page.



Feature 4: An administrator can create a link between products and suggestions. To make the amendment, administrator need to click on Product Suggestion Form. On clicking this button, the administrator will be routed to the below mentioned page



Feature 5: An administrator can check the Report of Most inactive users i.e who don't have any itinerary under their name. To check the report, administrator need to click on Inactive Users. On clicking this button, the below page will open.

List of Users without any itinerary

First Name	Last Name	emailID
Aditi	Chawla	adchawla@syr.edu
Aditya	Chauhan	achauh01@xyz.com
Ananya	B	ananya@gmail.com
Anmol	Handa	anmolhanda@gmail.com
Apoorva	Angre	aangre@gmail.com
Arjun	Suri	arsuri@gmail.com
Christiano	Ronaldo	cronaldo@gmail.com
John	Barnfield	jbarnfield@gmail.com
Katrina	Kaif	kkaif@gmail.com
Nidhi	Rakheja	nrakheja@gmail.com
Rajesh	A	rajesh@gmail.com
Roshan	Kumar	rkumar@gmail.com

Feature 6: An administrator can check the Report of Most active users i.e who have any itinerary under their name. To check the report, administrator need to click on Most Active Users. On clicking this button, the below page will open.

Most Active Users

First Name	Last Name	Number of Itinerary
Nitin	Khanna	1
Priyam	Shukla	2
Ranbeer	Kapoor	1
Niti	Saluja	1
Mandeep	Rana	1
Gauri	Gajaralwar	1
Zhihui	Hong	1
Sagar	Bhojwani	2

Sunday, December 3, 2017 Page 1 of 1

Feature 7: An administrator can check the Report of Most Preferred city among the users. To check the report, administrator need to click on Most Preferred Cities. On clicking this button, the below page will open.

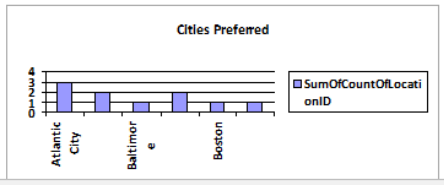
Screenshot 1:

Cities Preferred

City	Count
Atlantic City	3
Austin	2
Baltimore	1
Birmingham	2
Boston	1
Charlotte	1

Screenshot 2 :

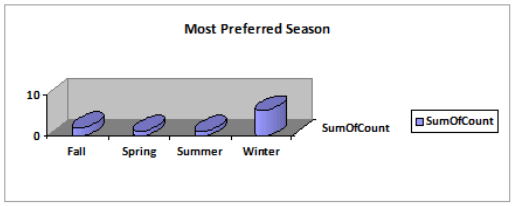
City	Count
Atlantic City	3
Austin	2
Baltimore	1
Birmingham	2
Boston	1
Charlotte	1



Feature 8: An administrator can check the Report of Most Preferred Season among the users. To check the report, administrator need to click on Most Preffered Season. On clicking this button, the below page will open.

Most Preferred Season

Season	Count
Fall	2
Spring	1
Summer	1
Winter	6

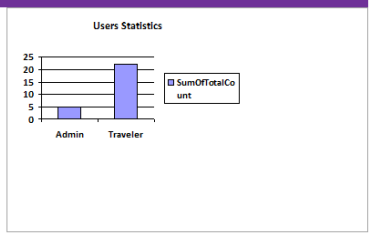


Sunday, December 3, 2017 Page 1 of 1

Feature 9: An administrator can check the also check the number of administrators and travelers in the system. The count gets updated automatically due to trigger running in the back end. The report also shows the graphical representation of the count in each category. To check the report, administrator need to click on Users Statistics. On clicking this button, the below page will open.

Users Statistics

User Type	User Description	TotalCount
R1	Admin	5
R2	Traveler	22



Trigger:

I implemented a trigger to update the count of Administrator and Traveler in the database

- ***Logic** : Whenever a new user is added to the UsersProject table, the count in the UserRoles table is updated according to the type of user (Administrator or Traveler)*

Trigger Code:

```
CREATE TRIGGER UsersCheck
ON UsersProject
FOR INSERT , UPDATE , DELETE
AS
IF
@@ROWCOUNT >= 1
BEGIN
    UPDATE UserRoles
    SET TotalCount = TotalUsers.total
    FROM
        (SELECT UserRoles.userDesc, COUNT( UsersProject.UserType) 'total'
        FROM UserRoles
        INNER JOIN UsersProject ON UserRoles.userType = UsersProject.UserType
        GROUP BY UserRoles.userDesc) AS TotalUsers
        WHERE UserRoles.userDesc=TotalUsers.userDesc
END;
```

Before the execution of the trigger:

```
SELECT * FROM UserRoles;
```

	userType	userDesc	TotalCount
1	R1	Admin	5
2	R2	Traveler	22

Inserting the Value in UsersProject

```
INSERT INTO UsersProject VALUES('Steven','McDonald',7283749858,'Male','smcdonald@gmail.com','Steven123','R2')
```

```
(2 row(s) affected)

(1 row(s) affected)
```

Checking the UserRoles Table again to verify whether Trigger has been implemented or not.

```
SELECT * FROM UserRoles;
```

	userType	userDesc	TotalCount
1	R1	Admin	5
2	R2	Traveler	23