

# Travel Planner

<https://github.com/andrewmlefebvre/TravelPlanner> <-- All source code

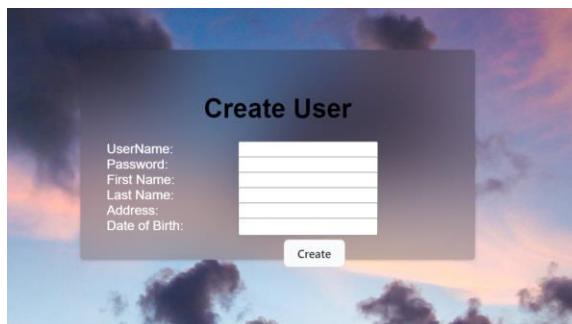
Andrew Lefebvre

## Project Idea

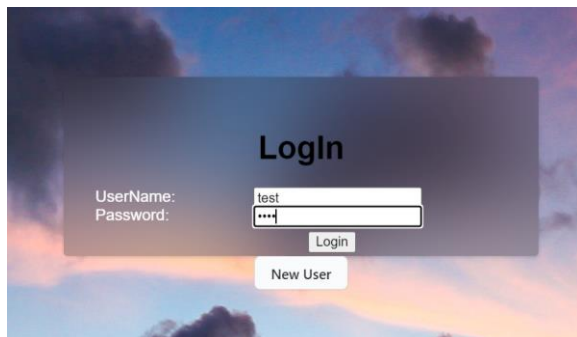
The idea of this project is to make a travel planner in which a user can plan a trip and receive information about the locations they are travelling to such as weather, nearby places and flight statuses.

## Basic Functions

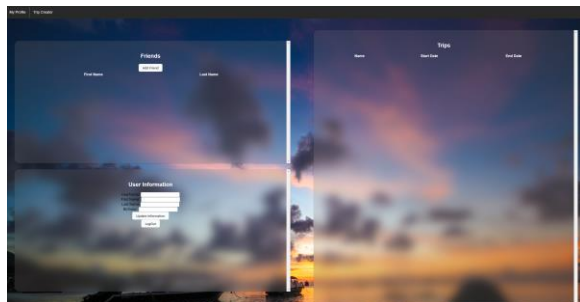
To use the application you start by creating an account

A screenshot of a web application's 'Create User' form. The form is centered on a background image of a sunset sky with clouds. It contains input fields for 'UserName:', 'Password:', 'First Name:', 'Last Name:', 'Address:', and 'Date of Birth:'. A 'Create' button is located at the bottom right of the form.

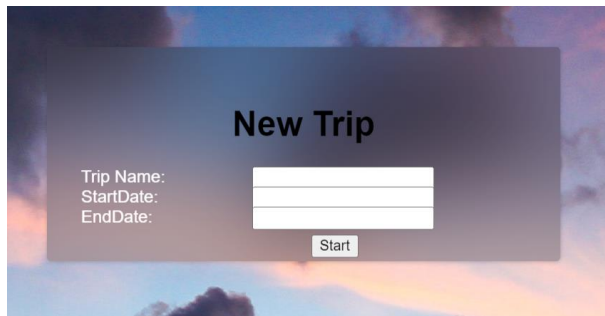
After creating, you log in on the front page

A screenshot of a web application's 'Login' form. The form is centered on a background image of a sunset sky with clouds. It contains input fields for 'UserName:' (with the text 'test' entered) and 'Password:' (with masked characters '\*\*\*\*' entered). There are 'Login' and 'New User' buttons at the bottom.

Once logged in you will see the main user interface of all the information



The user information can be changed on the bottom left corner of the screen. To create a trip you click in the Trip Creator tab

A screenshot of a 'New Trip' form. The form is titled 'New Trip' in bold black text. It contains three input fields: 'Trip Name:', 'StartDate:', and 'EndDate:'. Below these fields is a 'Start' button. The background of the form is a dark, semi-transparent rectangle over a sunset sky with clouds.

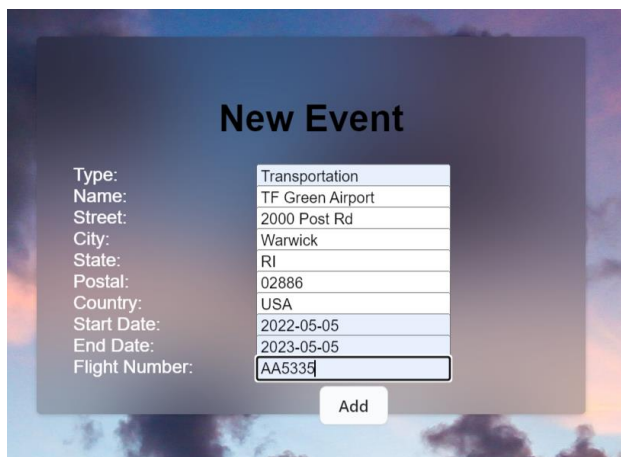
**New Trip**

Trip Name:

StartDate:

EndDate:

Once creating a trip you then add events

A screenshot of a 'New Event' form. The form is titled 'New Event' in bold black text. It contains several input fields: 'Type:', 'Name:', 'Street:', 'City:', 'State:', 'Postal:', 'Country:', 'Start Date:', 'End Date:', and 'Flight Number:'. Below these fields is an 'Add' button. The background of the form is a dark, semi-transparent rectangle over a sunset sky with clouds.

**New Event**

Type:

Name:

Street:

City:

State:

Postal:

Country:

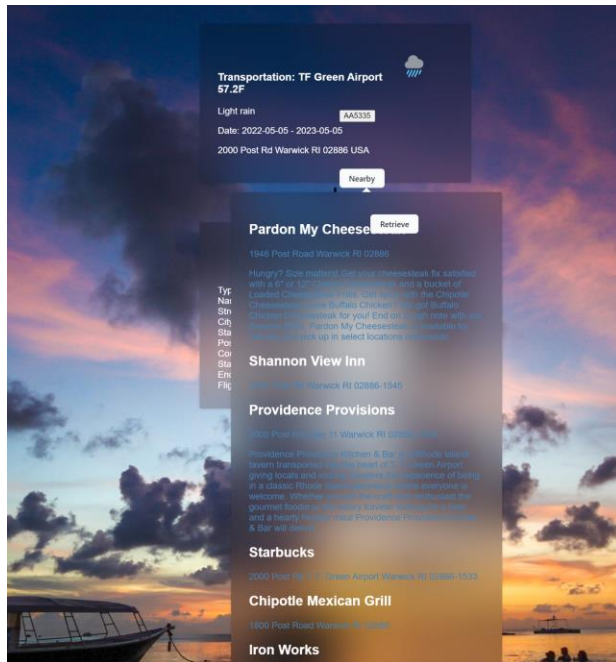
Start Date:

End Date:

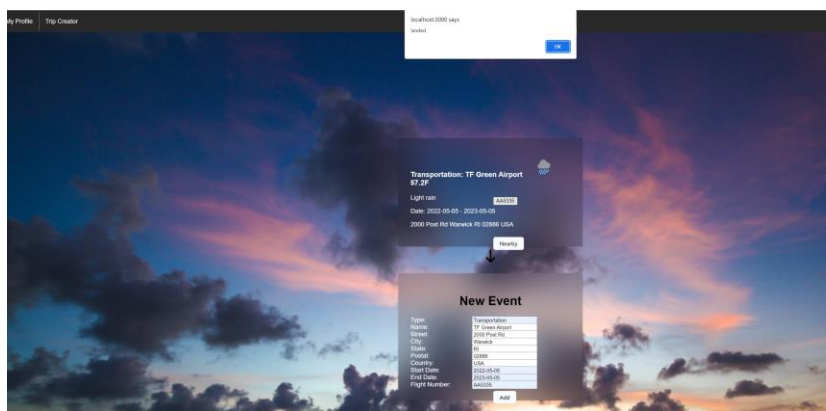
Flight Number:

The event will now appear with all of the information and current weather

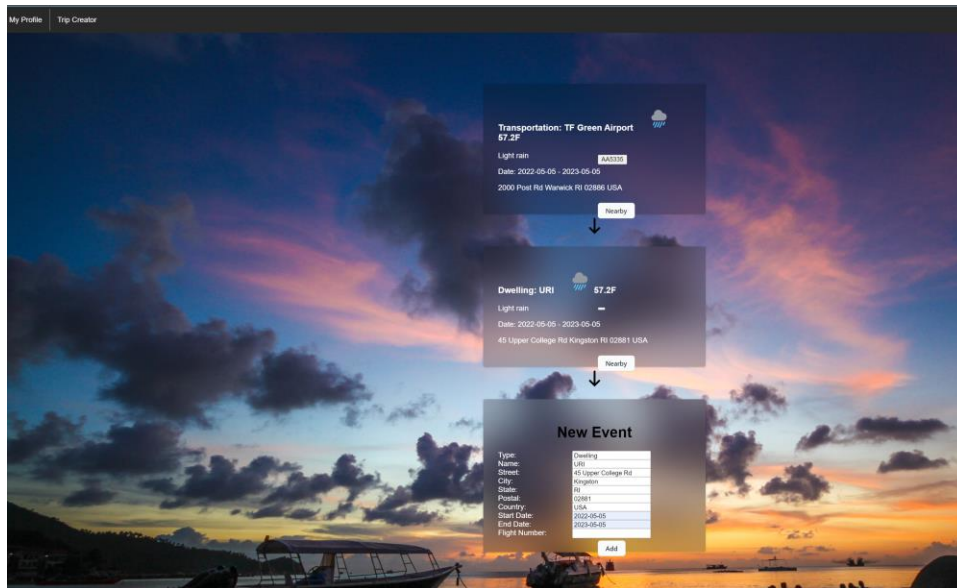
Clicking nearby will make an external API call to receive nearby places



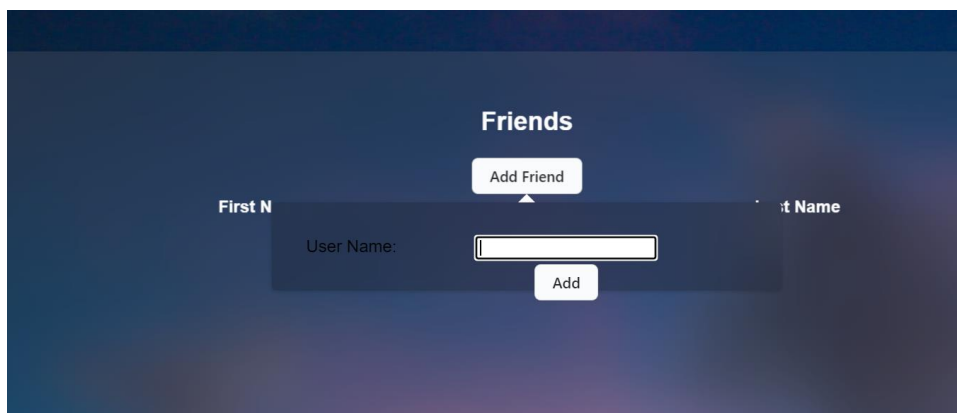
Clicking on the flight number will prompt an API call to check the status of the plane



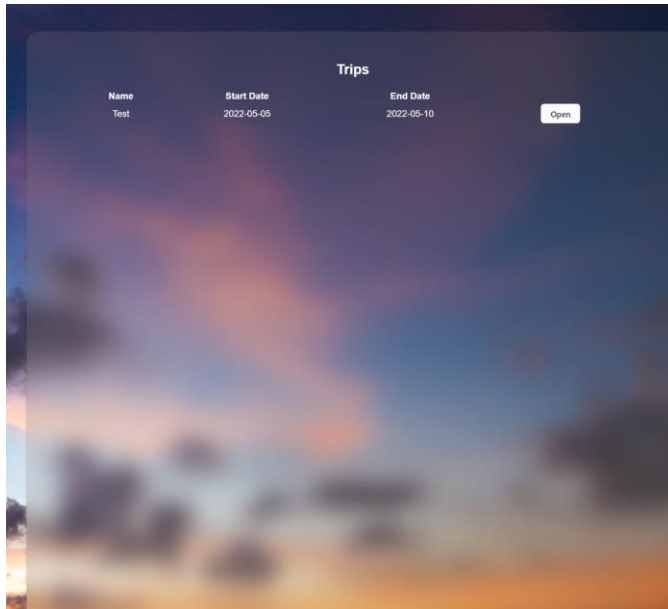
As many events can be added as required



On the user screen, you can add friends of other users

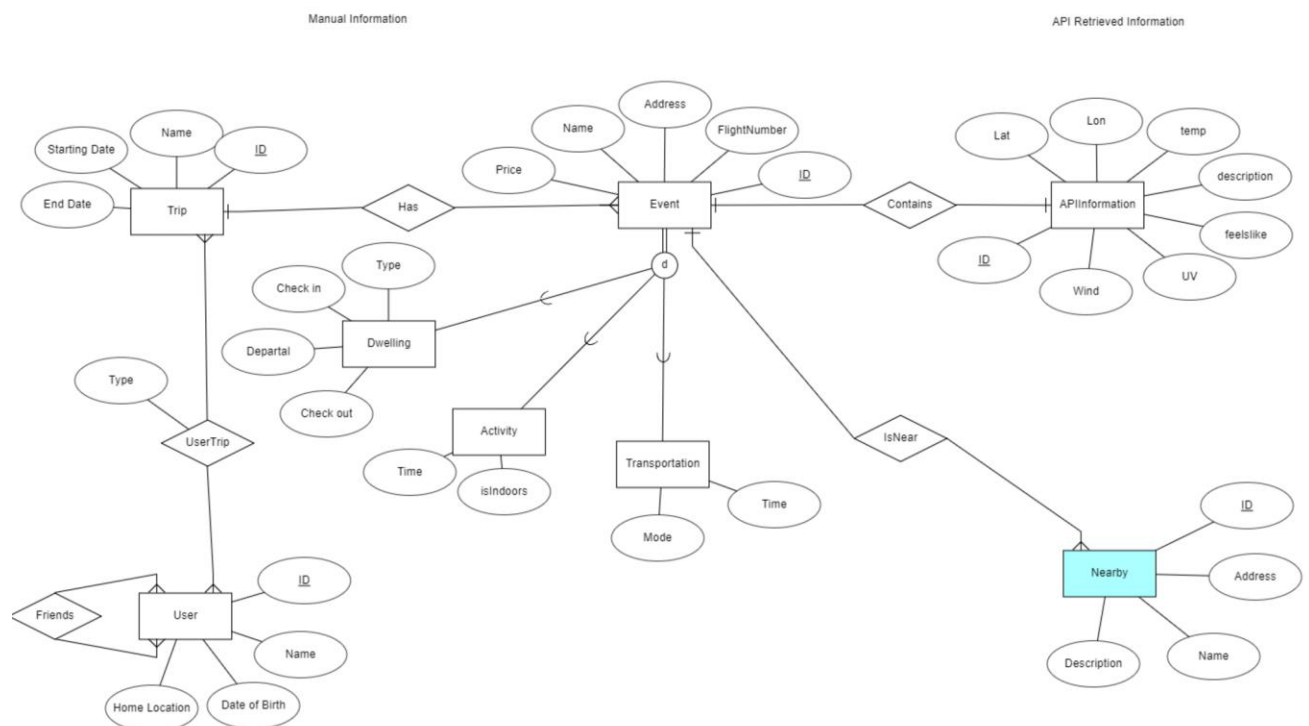


Once a friend is added all their trips become viewable under the trips list



Clicking open will bring you to the already created trip

## ERD



## API Sources

- Geocoding API (Turns address into coordinates)
- TripAdvisor API (Gives nearby places to address)

-Weather API (Gives weather for specific coordinates)

-Flight Status API

## Sample SQL Scripts

### Creating a User

```
INSERT INTO USER VALUES (null, 'User1', 'UserLastName1', '2000-5-3', '21 Test Ave Kingston RI 02882', 'User1', 'Password');
```

### Creating a Trip

```
INSERT INTO TRIP VALUES (null, 'Trip1', '2023-1-2', '2023-1-5');
```

### Adding a Friend

```
INSERT INTO FRIENDS VALUES (null, 1, 2);
```

### Selecting a friend

```
SELECT FRIENDS.friendID FROM FRIENDS JOIN USER on USER.ID = FRIENDS.friendID WHERE FRIENDS.userID = 1;
```



	friendID
▶	2
	3
	4

### Selecting a Trip from a User

```
SELECT TRIP.* FROM USER JOIN USERTRIP on USERTRIP.userID = USER.ID JOIN TRIP on TRIP.ID = USERTRIP.tripID WHERE USER.ID = 1;
```

	ID	name	startDate	endDate
▶	1	Trip1	2023-01-02	2023-01-05

Adding an event, and corresponding api information table

```
INSERT INTO EVENT VALUES (null,'Activity','Event1','45 Upper College RD','Kingston','RI','02881','USA',1,'2022-05-05','2022-05-05', null);
```

```
INSERT INTO APIINFORMATION VALUES (null, 1, null, null, null, null, null, null, null);
```

Selecting all existing events

```
SELECT * FROM Event;
```

	ID	subtype	name	street	city	state	postal	country	tripID	startDate	endDate	flightNumber
▶	1	Activity	Event1	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	2	Transportation	Transportation1	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	3	Activity	Event2	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	4	Transportation	Transportation2	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	5	Dwelling	Event3	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Deleting a Trip

```
DELETE trip FROM trip WHERE trip.ID = 1;
```

Selecting all events owned by a User across all trips

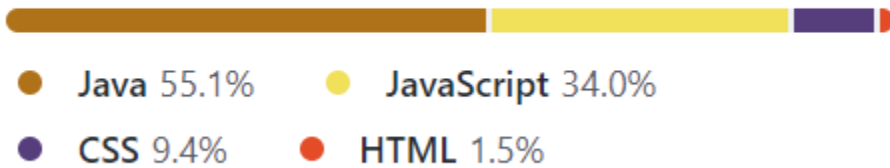
```
SELECT user.ID, event.subtype, COUNT(event.subtype) FROM USER LEFT JOIN USERTRIP on USER.ID = USERTRIP.userID LEFT JOIN TRIP on USERTRIP.tripID = trip.ID LEFT JOIN event on event.tripID = trip.ID GROUP BY User.ID, event.subtype ORDER BY USER.ID;
```

	ID	subtype	name	street	city	state	postal	country	tripID	startDate	endDate	flightNumber
▶	1	Activity	Event1	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	2	Transportation	Transportation1	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	3	Activity	Event2	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	4	Transportation	Transportation2	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
	5	Dwelling	Event3	45 Upper College RD	Kingston	RI	02881	USA	1	2022-05-05	2022-05-05	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Updating an API table

```
UPDATE APIINFORMATION SET temp = 66.0, des = 'Light rain', feelsLike = 66.0, UV = 1.0, wind = 2.2 WHERE eventID =1;
```

## Languages



Besides the initial sql code, most of this project leveraged a Spring REST server connecting with a REACT server. All API calls and SQL interactions were done in java, while all front-end code was written in react js.

Notable sections of code can be found in the git repositories README.md

Please view SampleSQLLogs.txt to see SQL actions taken during a simple web session

\*Because of the very limited amount of API calls available without buying the services this website will not be made available