



# Milestone 4 V2

**PREPARED FOR**

Jose Ortiz

**PREPARED BY**

Team 2

Globetrotter

JUL 30, 2021

Dear Jose Ortiz,

Please find enclosed our Milestone 4 document.

Yours Truly,

Team 2

# Globetrotter Travel Assistant

## Milestone 4

### Team 2

Role	Name	Email
Team Lead/Github Master	Taylor Artunian	tartunian@mail.sfsu.edu
Front-End Lead	Kajeme Cheneque	
Back-End Lead	Luis Alfaro	
Front End	Ruja Rajbhandari	
	Yi Liu	
	Jesus Correa	
	Robin Fernando	

### Revision History

Version	Date Submitted	Date Revised
M4V1	07/30/21	08/02/21
M3V2	07/30/21	-
M3V1	07/22/21	07/30/21
M2V2	07/22/21	-
M2V1	07/08/21	07/22/21
M1V2	07/08/21	-
M1V1	06/22/21	07/08/21

# Table of Contents

	1
<b>Table of Contents</b>	<b>4</b>
<b>1. Product Summary</b>	<b>6</b>
Priority 1 Functional Requirements	6
Account Management	6
Login	6
Photo Album	7
Route Planner	7
Saved Trips	7
Checkout	7
Previous Trips	7
Registration	7
Meet the Team	8
Website	8
<b>2. Usability Test Plan</b>	<b>9</b>
Outline	9
Objectives	9
Test Description	9
Testing Steps	10
Search for a flight	10
View a previous trip - KJ	10
Reopen a saved trip - Ruza	10
Upload an image - Jesus	10
Learn about the team members - Yi	10
Questionnaire	11
Usability Testing Results	12
<b>3. QA Test Plan</b>	<b>13</b>
<b>4. Code Review</b>	<b>15</b>
Coding Style	15
Email Correspondence	16
Code & Comments From Attached Document	17

<b>5. Self-Check on Best Practices</b>	<b>20</b>
Images	20
Passwords	21
Input Validation	22
Registration	22
Route Planner	23
<b>6. Non-Functional Requirements</b>	<b>25</b>
<b>7. Contributions</b>	<b>31</b>

# 1. Product Summary

Product Name: Globetrotter Travel Assistant

At Globetrotter, we are aware that the internet has endless opportunities. That being said, we plan to use the internet as our main source of marketing. One way we plan to market is through Youtube. I'm sure we have all seen those ads that youtubers put into their videos. The beauty of those ads is that a viewer must watch a certain amount of the ad before they can skip it. Getting an ad, with information about our product, to play in between videos would ensure that people will see it. We could also make a deal with youtubers that have a huge audience and have them talk about our product at the beginning of their video. Millions of viewers, of all ages, would see their favorite content creators talk about our product. This is great for our product because youtubers with huge followings can easily persuade their fans.

## Priority 1 Functional Requirements

### Account Management

1. Registered users can edit their birthday.
2. Registered users can edit their email.
3. Registered users can edit their home city
4. Registered users can edit their home postal code.
5. Registered users can edit their home state
6. Registered users can edit their home street address.
7. Registered users can edit their password.
8. Registered users can edit their primary phone number.
9. Registered users can edit their profile photo.
10. Registered users can edit their secondary phone number.
11. Registered users can edit their username.
12. Registered users can view the 'Account Management' page.
13. Unregistered users cannot access Account Management.

### Login

14. Registered users can login using the "Login" page.
15. Unregistered users can view the "Registration" page.
16. Users shall provide their password when logging in.
17. Users shall provide their username when logging in.

18. Registered users will be able to reset their password by providing their username, email, and new password.
19. The password will be obscured in the Login page.

### **Photo Album**

20. Registered users can upload pictures of their trip.
21. Unregistered users cannot access Photo Albums.

### **Route Planner**

22. All registered users can add at least one destination to their trip..
23. All registered users can add multiple Activities to a Trip.
24. All registered users can edit the destination of a trip.
25. All registered users can edit the starting location of a trip.
26. All registered users can remove the destination of a trip.
27. All registered users can remove the starting location of a trip.
28. All registered users can remove Activities from a Trip.
29. All registered users shall specify the dates of departure for each flight.
30. Unregistered users cannot access the Route Planner.

### **Saved Trips**

31. Registered users can view trips that they have saved.
32. Unregistered users cannot access Saved Trips.

### **Checkout**

33. Unregistered users cannot access Checkout.

### **Previous Trips**

34. Unregistered users cannot access Previous Trips.

### **Registration**

35. Unregistered Users shall provide their date of birth when creating an account
36. Unregistered users shall provide a password when creating an account.
37. Unregistered users shall provide a primary phone number when creating an account.
38. Unregistered users shall provide a username when creating an account.
39. Unregistered users shall provide their email when creating an account.

- 40. Unregistered users shall provide their first name when creating an account.
- 41. Unregistered users shall provide their home city when creating an account.
- 42. Unregistered users shall provide their home postal code when creating an account.
- 43. Unregistered users shall provide their home state when creating an account.
- 44. Unregistered users shall provide their home street address when creating an account.
- 45. Unregistered users shall provide their last name when creating an account.
- 46. Unregistered users will be able to create an account.
- 47. Unregistered users shall provide their gender when creating an account.

### **Meet the Team**

- 48. All users can contact the team.
- 49. All users can view the “About Us” page to find out more information about the team.

### **Website**

- 50. The website will be easy for all Users to maneuver through.

Our product is unique in that it has a photo gallery feature. Based on our research, no website that allows users to book flights has a photo gallery feature. In our website, once a user has created and purchased a trip, they can go back and upload images to previous trips they have been on. This benefits the user because they have a place where they can store some of their favorite images that they took while on that specific trip. Some of our future plans for this feature include being able to invite guests that went on the trip with the owner, to the photo gallery so that both the owner and the guests can upload images. We decided to go with this unique feature because a lot of us can agree that memories are priceless and we wanted to give the users the option to save any memories they would like to keep forever.

**Visit our Home Page At:** <http://172.93.54.146:3000/>



## 2. Usability Test Plan

### Outline

#### Objectives

In these usability tests, our goal is to assess whether the various parts of our website are easy to find and use. The five parts or features that we have determined should be tested include: *searching for a flight*, *viewing a previous trip*, *reopening a saved trip*, *uploading an image*, and also *learning about the team members*. In checking the flight search feature, we want to ensure that users can easily find the input fields, that the airport autocomplete is easy to use, and that the flight results show enough information. With viewing a previous trip, we want to know that a user can find the Previous Trips page and find a specific trip that they have taken. From here we will test the usability of the Photo Gallery feature. The test will determine how easy it is to find the Photo Gallery for a particular trip as well as if there is any difficulty in uploading an image. Next, we will have the user reopen a saved trip. Here, we will test whether the user can find the Saved Trips page, view the details of the saved trip, and figure out how to reopen the saved trip. Lastly, we will assess how accessible the Meet the Team pages are, how easily a user can access them, and also find information on each team member.

#### Test Description

To perform the usability tests, there are not very many requirements. The user should use a desktop or laptop computer with internet access to perform the steps listed below. When performing the test, the user does not need to return home or start from a specific page in order to move on to the next test.

To begin the test users should navigate to our website at: <http://172.93.54.146:3000/login>

At the login page, users may use the built-in account  
username: **globetrotter**  
password: **globetrotter**

## Testing Steps

### Save a Trip

1. The user will first attempt to navigate to the Route Planner.
2. The user will attempt to input the airport from which they would like to depart and select from the autocomplete results.
3. The user will attempt to input the airport to which they would like to travel and select from the autocomplete results.
4. The user will attempt to select the desired date of departure.
5. The user will attempt to search for flights matching the parameters.
6. The user will wait for the flight results to load.
7. The user will view the results and decide which result is best for them.
8. The user may add additional activities using the 'Add Activity' button and repeat steps 2 through 7.

1	Task	Description
	Task	Save a Trip
	Machine State	In the Route Planner.
	Successful Completion	Trip will show up in the "Saved Trips" page.
	Benchmark	30 seconds

### View a previous trip - KJ

1. The user will attempt to navigate to the Previous Trips page.
2. The user will attempt to find a Previous saved Trip.
3. The user will click the file picker button.
4. The user will select a file from their computer.
5. The user will click the upload image button.

2	Task	Description
	Task	View a previous trip.
	Machine State	On the home page, already logged in.
	Successful Completion	The user will be able to see previously created trips.
	Benchmark	20 seconds

#### Upload an image - Jesus

1. The user will attempt to find the Previous Trips page.
2. The user will have to click on the photo gallery option of the previous trip.
3. The user will have to click the "Choose a File" button.
4. The user will then have to choose the image they would like to upload.
5. The user will click the "Upload" button to upload the image.

3	Task	Description
	Task	Upload an image.
	Machine State	On the home page, already logged in.
	Successful Completion	The photo gallery will have the users uploaded photos.
	Benchmark	60 seconds

#### Reopen a saved trip - Ruza

1. The user will attempt to find the Saved Trips page.
2. The user will attempt to open a specific saved trip.
3. The user will view the details of the saved trip.

4	Task	Description
	Task	Reopen a saved trip.
	Machine State	On the home page, already logged in.
	Successful Completion	The user will be in the route planner, with the details of the reopened trip on the screen.
	Benchmark	30 seconds

#### Learn about the team members - Yi

1. The user will attempt to find the About page.
2. The user will have to click on a member's photo or name to get the member's details.
3. The user will attempt to access each member's information page.

5	Task	Description
	Task	Learn about the team members.
	Machine State	On the home page.
	Successful Completion	The user will be reading information about one of the team members from their individual "About Me" page.
	Benchmark	60 seconds

## Questionnaire

For each of these questions, please rate your response as one of the following:

*Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree*

1. It is easy to find the list of activities in the Route Planner.
2. The airport search feature was easy to use.
3. The flight results show enough information for me to pick a flight in the Route Planner.
4. It is easy to find the Previous Trips page.
5. It is easy to see previous saved trips.
6. It is easy to find all previous trips.
7. It is easy to click the file picker button.
8. It is easy to select an image to upload from the computer.
9. It is easy to click on the upload image button.
10. It is easy to find the Saved Trips page.
11. It is easy to open a specific saved trip.
12. It is easy to find details of the saved trip.
13. The process of uploading an image is simple.
14. It is easy to find the photo gallery.
15. I was able to upload more than one image.
16. It is easy to find the About page.
17. It is easy to find all team members' photos and names.
18. It is easy to find a specific team member's detailed information.

## Questionnaire Results

Question #	Participant								
	1	2	3	4	5	6	7	8	9
1	Agree	Agree	Agree	Agree	Agree	Neutral	Neutral	Agree	Agree
2	Agree	Strongly Agree	Agree	Agree	Agree	Strongly Agree	Neutral	Agree	Neutral
3	Neutral	Strongly Agree	Disagree	Neutral	Neutral	Strongly Agree	Agree	Strongly Agree	Disagree
4	Agree	Strongly Agree	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree
5	Neutral	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree
6	Neutral	Agree	Neutral	Neutral	Agree	Strongly Agree	Agree	Agree	Neutral
7	Agree	Agree	Agree	Agree	Neutral	Agree	Agree	Agree	Agree
8	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree
9	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree
10	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Disagree	Agree
11	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree
12	Neutral	Agree	Agree	Agree	Agree	Strongly	Agree	Agree	Agree

						Agree			
<b>13</b>	Disagree	Agree	Agree	Agree	Neutral	Strongly Agree	Agree	Agree	Agree
<b>14</b>	Neutral	Agree	Agree	Neutral	Neutral	Strongly Agree	Agree	Agree	Agree
<b>15</b>	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree
<b>16</b>	Neutral	Neutral	Strongly Disagree	Neutral	Neutral	Strongly Agree	Strongly Disagree	Disagree	Neutral
<b>17</b>	Disagree	Disagree	Strongly Disagree	Neutral	Neutral	Strongly Disagree	Strongly Disagree	Disagree	Disagree
<b>18</b>	Neutral	Disagree	Strongly Disagree	Neutral	Neutral	Strongly Disagree	Strongly Disagree	Disagree	Disagree

## Usability Testing Results

Test Name	% Completed	Errors	Comments	% Time	Response Distribution*												
Save a trip	90%	Some UI elements glitched	Date is missing from flight results. Saving does not work.	30s	<table><thead><tr><th>Response</th><th>Count</th></tr></thead><tbody><tr><td>Strongly Disagree</td><td>1</td></tr><tr><td>Disagree</td><td>1</td></tr><tr><td>Neutral</td><td>1</td></tr><tr><td>Agree</td><td>12</td></tr><tr><td>Strongly Agree</td><td>4</td></tr></tbody></table>	Response	Count	Strongly Disagree	1	Disagree	1	Neutral	1	Agree	12	Strongly Agree	4
Response	Count																
Strongly Disagree	1																
Disagree	1																
Neutral	1																
Agree	12																
Strongly Agree	4																
View a previous trip	100%	None.	Only one trip available to view	20s	<table><thead><tr><th>Response</th><th>Count</th></tr></thead><tbody><tr><td>Strongly Disagree</td><td>1</td></tr><tr><td>Disagree</td><td>1</td></tr><tr><td>Neutral</td><td>1</td></tr><tr><td>Agree</td><td>32</td></tr><tr><td>Strongly Agree</td><td>8</td></tr></tbody></table>	Response	Count	Strongly Disagree	1	Disagree	1	Neutral	1	Agree	32	Strongly Agree	8
Response	Count																
Strongly Disagree	1																
Disagree	1																
Neutral	1																
Agree	32																
Strongly Agree	8																
Reopen a saved trip	100%	None.	No saved details	30s	<table><thead><tr><th>Response</th><th>Count</th></tr></thead><tbody><tr><td>Strongly Disagree</td><td>1</td></tr><tr><td>Disagree</td><td>1</td></tr><tr><td>Neutral</td><td>1</td></tr><tr><td>Agree</td><td>18</td></tr><tr><td>Strongly Agree</td><td>2</td></tr></tbody></table>	Response	Count	Strongly Disagree	1	Disagree	1	Neutral	1	Agree	18	Strongly Agree	2
Response	Count																
Strongly Disagree	1																
Disagree	1																
Neutral	1																
Agree	18																
Strongly Agree	2																
Upload an image	100%	None.	Should be accessible elsewhere	60s	<table><thead><tr><th>Response</th><th>Count</th></tr></thead><tbody><tr><td>Strongly Disagree</td><td>1</td></tr><tr><td>Disagree</td><td>1</td></tr><tr><td>Neutral</td><td>1</td></tr><tr><td>Agree</td><td>15</td></tr><tr><td>Strongly Agree</td><td>2</td></tr></tbody></table>	Response	Count	Strongly Disagree	1	Disagree	1	Neutral	1	Agree	15	Strongly Agree	2
Response	Count																
Strongly Disagree	1																
Disagree	1																
Neutral	1																
Agree	15																
Strongly Agree	2																
Learn about the team members	30%	Broken link	Have to use another port	60s	<table><thead><tr><th>Response</th><th>Count</th></tr></thead><tbody><tr><td>Strongly Disagree</td><td>8</td></tr><tr><td>Disagree</td><td>8</td></tr><tr><td>Neutral</td><td>10</td></tr><tr><td>Agree</td><td>0</td></tr><tr><td>Strongly Agree</td><td>1</td></tr></tbody></table>	Response	Count	Strongly Disagree	8	Disagree	8	Neutral	10	Agree	0	Strongly Agree	1
Response	Count																
Strongly Disagree	8																
Disagree	8																
Neutral	10																
Agree	0																
Strongly Agree	1																

\* Charts include combined responses for all questionnaire questions belonging to that Test (blue=strongly disagree, red=disagree, orange=neutral, green=agree, light red=strongly agree)



### 3. QA Test Plan

#### QA Testing Steps

**Test Name: Secured Password in Registration**

**Purpose: See if the user's password is encrypted in database**

Step #	Step Description
1	Fill out the registration form and submit (or send a POST request to /users/register).
2	Search the database for registered users (SELECT username, password_hashed FROM registered_user).
3	Check if the password_hashed column contains a cleartext password or a hash value.

**Test Name: User Sessions at Login**

**Purpose: Check if a user session is created after login**

Step #	Step Description
1	Fill out the login form and submit.
2	Check the right-most part of the navbar for the logged in user's username.

**Test Name: Flight Searching via Amadeus API****Purpose: To see if correct flights are retrieved based on parameters**

Step #	Step Description
1	Make a GET request to our flight search endpoint: /planner/flights?originLocationCode=LAX&destinationLocationCode=SFO&departureDate=2021-09-02&adults=1&currencyCode=USD&max=10
2	Change the query parameters to match the desired test.
3	Continue to the page.
4	Ensure that the results returned match the parameters inputted in the previous step.

**Test Name: Check Access Control to Pages****Purpose: To make sure that pages which require login redirect anonymous users to the login page**



Step #	Step Description
1	Go to the website and logout if already logged in.
2	Try navigating to a restricted page such as /planner or /accountmanagement.
3	Ensure that you are redirected to the login page.

**Test Name: No Location Restriction**

**Purpose: Make sure that the website is accessible from other countries.**

Step #	Step Description
1	Send the website link to someone outside of the US (or use a VPN/proxy to access the website).
2	Check that the website loads under these conditions.

## QA Test Results

#	Test	Purpose	Input/User Action	Expected Result	Pass/Fail
1	Secured password in registration	See if the user's password is encrypted in database	Registration post request to /users/register	The password is encrypted before sending to database	Pass
2	User sessions at login	Check if a user session is created after login	Successful authentication	The user's username will be pulled from the session data and displayed in the navbar	Pass
3	Flight searching via Amadeus API	To see if correct flights are retrieved based on parameters	<pre>var response = await amadeus.shopping.flightOffersSearch.get({   originLocationCode: 'LAX',   destinationLocationCode: 'SFO',   departureDate: '2021-08-04',   currencyCode: 'USD',   adults: 1,   nonStop: true,   max: 5 });</pre>	Results should be displayed like this: 0 LAX to SFO  PT1H23M USD 38.40 1 LAX to SFO  PT1H26M USD 38.40	Pass

4	Access Control to Certain Pages	Check if expected pages are inaccessible to users that are not logged in	Manually visiting /savedtrips	Redirect to /login	Fail
5	No Location Restriction	To ensure users can access the website outside of the US	Visit our URL <a href="http://172.93.54.146:3000/">http://172.93.54.146:3000/</a> from Iceland	The home page should display	Pass

## **4. Code Review**

### Coding Style

Capitalization: camelCase

Indentation Style: “One True Brace” - braces go on the same line as function declarations, if-else, etc.

*(email correspondence begins on next page)*

## Email Correspondence

**From:** Taylor Artunian <[tartunian@mail.sfsu.edu](mailto:tartunian@mail.sfsu.edu)>  
**Sent:** Wednesday, July 28, 2021 10:13 PM  
**To:** Luis Adrian Alfaro <[lalfaro4@mail.sfsu.edu](mailto:lalfaro4@mail.sfsu.edu)>  
**Subject:** CSC648 Team2 Code Review

Hey Luis,

I am submitting a portion of my code from the Route Planner to you so that you can perform a code review. Attached is the client-side JavaScript for the Route Planner. Please comment on any part of the code, check for consistency with our coding style, and share your feedback with me when you have some time.

Thanks,

Taylor

**From:** Luis Adrian Alfaro <[lalfaro4@mail.sfsu.edu](mailto:lalfaro4@mail.sfsu.edu)>  
**Sent:** Thursday, July 29, 2021 2:34 PM  
**To:** Taylor Artunian <[tartunian@mail.sfsu.edu](mailto:tartunian@mail.sfsu.edu)>  
**Subject:** Re: CSC648 Team2 Code Review

Good Afternoon Taylor,

Good job on following our coding standards in the code you submitted for code review. I did notice some issues, most of them being comments, and have highlighted my comments regarding those issues. In the attached document you shall see my comments around the parts where I did notice some problems. Other than that everything else was good!

Best Regards,

Luis Alfaro

## Code & Comments From Attached Document

//No comment explaining the fetchURL function. -L.A.

```
async function fetchURL(method, endpoint, parameters) {  
    var url = new URL(getURLBase() + endpoint),  
        params = parameters  
    Object.keys(params).forEach(key => url.searchParams.append(key, params[key]));  
    var response = await fetch(url, {  
        method: method,  
        headers: new Headers({  
            // 'Authorization': `Bearer ${token}`,
```

//Will new Headers always be empty, since there is commented out code inside of new Headers? -L.A.

```
    });  
    });  
    return response;  
}
```

/\*\*\*\*\*

\* Event handler for planner-result-item-button when it is within an Activity

//Comment not describing what this event handler is going to do, only describes a  
//condition. -L.A.

\*\*\*\*\*/

```
function removeFlightClickHandler(e) {  
    // Get the parent result item of the button  
    var resultItem = e.target.closest('.planner-result-item');
```



```

    resultItem.remove();
}

/*****

* Event handler for planner-result-item-button when it is in the result list

//Comment not describing what this event handler is going to do, only describes a
//condition. -L.A.

*****/

function addFlightClickHandler(e) {

    // Get the parent result item of the button

    var resultItem = e.target.closest('.planner-result-item');

    // Get the currently selected Activity and child container

    var selectedActivity =
document.querySelector('input[name="planner-selected-activity"]:checked').parentNode.pa
rentNode;

    var selectedActivityChildContainer =
selectedActivity.querySelector('.planner-activity-child-container');

    // Get the results container

    var resultsList = document.getElementById('planner-results-container');

    // Make a copy of the flight offer

    var copyOfFlightOffer = resultItem.cloneNode(true);

    var flightOfferButton = copyOfFlightOffer.querySelector('.planner-result-item-button')

```

```
flightOfferButton.value = 'Reset';

flightOfferButton.addEventListener('click', removeFlightClickHandler);

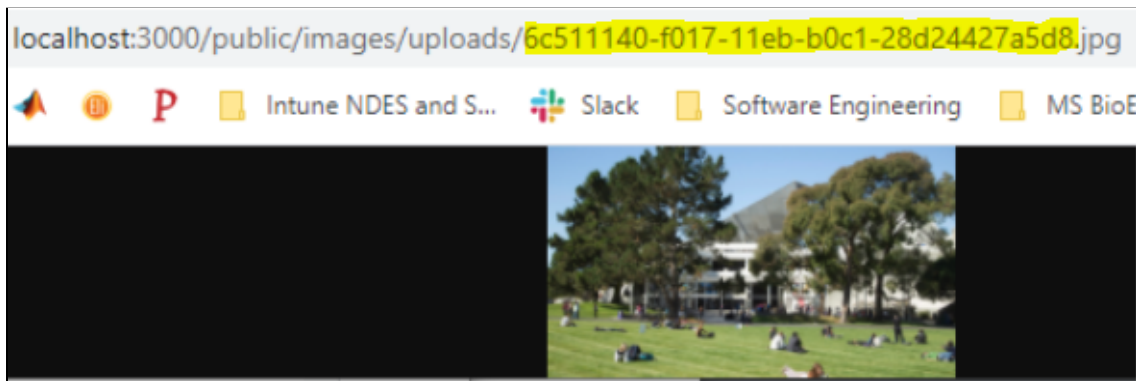
//typo in under -L.A

// Move the result item (new flight offer) under the selected Activity
if (selectedActivityChildContainer.childElementCount == 0) {
    selectedActivityChildContainer.innerHTML = "";
    selectedActivityChildContainer.appendChild(copyOfFlightOffer);
} else {
    selectedActivityChildContainer.replaceChild(copyOfFlightOffer,
selectedActivityChildContainer.children[0]);
}
}
```

## 5. Self-Check on Best Practices

### Images

When photos are uploaded to our website, random UUID's are used for file names. This is done to avoid revealing a user's name or other personal information which may be in the original file name.



Additionally, we are protecting the static images of the website which should not be publicly accessible. If a user is not logged in, they can no longer access photos uploaded to the Photo Gallery even if they manually enter the URL.

Message: You must log in to access /6c511140-f017-11eb-b0c1-28d24427a5d8.jpg

### Globetrotter

Username

Password:

[Forgot Password?](#)

By resetting your password you agree to the [Terms of Use](#) and [Privacy Policy](#)

Login

Dont have an account? [Sign Up](#)

## Passwords

We hash the password using the `.hash()` function from the `bcrypt` library before storing into the database. As you can see in the `createUser()` function, line 147, we hash the password. We store the hashed password into a variable and pass it into an array that will be used as a parameter for the query. We then store the password on line 153 and at no point does the Globetrotter team know what the password of the newly created user is.

```
139
140 /*****
141  * Create a user in the database (all fields)
142  *****/
143 async function createUser(email, username, password, firstName, lastName, birthday, gender,
144     preferredCurrency, homeLocationAddressLine1, homeLocationAddressLine2,
145     homeLocationCity, homeLocationState, homeLocationCountry, homeLocationPostalCode,
146     primaryPhoneCountryCode, primaryPhoneNumber) {
147     var passwordHash = await bcrypt.hash(password, 10);
148     var query = 'CALL usp_register_user(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, @user_id)';
149     var params = [email, username, passwordHash, firstName, lastName, birthday, gender,
150         preferredCurrency, homeLocationAddressLine1, homeLocationAddressLine2,
151         homeLocationCity, homeLocationState, homeLocationCountry, homeLocationPostalCode,
152         primaryPhoneCountryCode, primaryPhoneNumber];
153     var result = await runQuery(query, params);
154     return result;
155 }
```

```
+-----+-----+
| username | password_hashed |
+-----+-----+
| globetrotter | $2b$10$4TF8lwcJBfFW.tgew44ub0fQQCR7oZNxAg.UJXI.zwgdW5xMZd6US |
| GLOBETROTTER | $9b$99$9TF8lwcJBfFW.ftqd44ub0fQQCR7oZNxAg.UJXI.zwgdW5xMZd9EU |
+-----+-----+
2 rows in set (0.06 sec)

mysql>
```

# Input Validation

## Registration

All of the fields in our registration form are specified as required with the *required* HTML tag.

By doing this, we reduce the number of calls to our registration endpoints and prevent needless page reloads for the user.

When a user submits the registration form, we check our database first to see if the username or email are already in use.

```
// Check if username is already in use
var user = await database.getUserByUsername(username);
if (user) {
  log('username already in use', 'fail');
  return res.redirect(`/registration?message=username already in use.`);
}
```

```
// Check if email is already in use
user = await database.getUserByEmail(email);
if (user) {
  log('email already in use', 'fail');
  return res.redirect(`/registration?message=email already in use.`);
}
```

```
<form method="POST" action="/users/register" class="center">
  <h1>Create an account to access Globetrotter</h1>
  <div>{{message}}</div>
  <p class="light-text">Fill the form below to create an account with Globetrotter</p>

  <div class="row">
    <label>First Name:</label>
    <input type="text" name="firstName" placeholder=" Jane" required>
  </div>
  <div class="row">
    <label>Last Name:</label>
    <input type="text" name="lastName" placeholder=" Doe" required>
  </div>
```

## Create an account to access Globetrotter

username already in use.

Fill the form below to create an account with Globetrotter

First Name:

Last Name:

This is the Registration page. If the registration is unsuccessful, we display a message to the user explaining why.

## Route Planner

Before a user can search for flights, all three inputs must be valid. The code below checks whether the fields are filled in, whether the departure date is valid, and whether the airport codes are valid options from the autocomplete list.

```
-----Excerpt from planner.js-----
if(isIncompleteActivity(activity)) {
  alert('Some search parameters are invalid.');
```

return;

```
}
if (!validateAirportInput(originInputValue, originInputDatalist) ||
!validateAirportInput(destinationInputValue, destinationInputDatalist)) {
  alert('Please input a valid airport.');
```

return;

```
}

-----Excerpt from validator.js-----
function validateAirportInput(inputValue, datalistElement) {
  for (var option of datalistElement.options) {
    var airportCode = option.value;
    if (inputValue == airportCode) {
      return true;
    }
  }
  return false;
}

function isIncompleteActivity(activity) {
  var originCode = activity.querySelector('.planner-activity-origin-input').value;
  var destinationCode =
activity.querySelector('.planner-activity-destination-input').value;
  var departureDate = activity.querySelector('.planner-activity-departure-date').value;
  if (originCode == "" || destinationCode == "" || isNaN(Date.parse(departureDate))) {
    return true;
  }
  return false;
}
```

Best Trip Ever (2021) Flight Search Results

SFO to JFK

08/29/2021

Search Remove

0

SFO to JFK

10:25 PM 07:00 AM

jetBlue

PT5H35M

USD 145.40

Reset

Trip Name

Destination Airport Code

Origin Airport Code

Departure Date

Search Button

This is the Route Planner. The code files referenced are client-side JavaScript code used on this page. “*planner.js*” is the main script for loading the data from the server and displaying the UI elements. “*validator.js*” is a separate script with just the two functions shown and is used to validate the UI elements/inputs.

When the user saves the Trip, the same code as above is run, but this time for all Activities in the Trip. Additionally, the Trip Name field is validated for empty values.

```
if(tripName=="") {  
    alert('Please enter a valid name for your trip.');
```

```
    return;
```

```
}
```

## 6. Non-Functional Requirements

Category	Non-Functional Requirement	Status
Functionality	The website shall be hosted on SSD Nodes.	DONE
Functionality	The website shall be user friendly.	DONE
Functionality	The website shall be easy to navigate.	DONE
Security	User login credentials will be validated against the user database.	DONE
Security	Changes to access controls in the database will be reflected immediately in the user interface.	DONE
Security	The user's passwords will be encrypted in the database.	DONE
Security	User passwords will be case sensitive.	DONE
Security	User passwords shall have at least 8 characters.	DONE
Security	User passwords shall have at least one upper case letter.	DONE
Security	User passwords shall have at least one special character.	DONE
Security	The website will use sessions to track logged in users.	DONE
System Requirements	The website shall function on at least version 91.0.4472.101 of Google Chrome.	DONE
System Requirements	The website shall function on at least version 91.0.864.48 of Microsoft Edge.	DONE
System Requirements	The website shall function on at least version 14.1 of Apple Safari.	DONE
System Requirements	The website shall function on at least version 89.0.1 of Mozilla Firefox.	DONE
Marketing	The website shall use the company logo displayed.	DONE



Marketing	The company shall have engaging content.	DONE
Content	The website shall have a navigation bar.	DONE
Content	The website shall have a consistency of its design on every page.	ON TRACK
Content	The flights returned from API will match parameters entered by the user.	DONE
Coding Standards	The code will be portable.	DONE
Coding Standards	The code will have a consistent indentation style.	ON TRACK
Coding Standards	The code shall be understandable.	ON TRACK
Coding Standards	The code shall contain comments so that team members can quickly learn what different pieces of code do without having to dive deep into the code.	ON TRACK
Coding Standards	There shall be a limit to the line length of the code to avoid horizontal scrolling.	ON TRACK
Coding Standards	The team lead shall decide if code is at the agreed upon standard.	ON TRACK
Coding Standards	The team lead shall decide whether code can be pushed to the main branch.	DONE
Coding Standards	The coding will be done in a modular way.	DONE
Coding Standards	Code modules will be reused as often as possible.	DONE
Coding Standards	CSS will be applied to classes.	DONE
Coding Standards	Scripts will be in their own files	DONE
Coding Standards	NodeJS modules will be installed in the 'node_modules' folder.	DONE

Coding Standards	The website will use a public and private folder to serve web content.	DONE
Coding Standards	Stylesheets will be in their own files.	DONE
Coding Standards	HTML files will be in the HTML folder.	DONE
Coding Standards	CSS files will be in the CSS folder.	DONE
Coding Standards	Script files will be in the Script folder.	DONE
Coding Standards	Express routers will be placed in their own folder.	DONE
Coding Standards	Handlebars partials will be in their own folder.	DONE
Coding Standards	Handlebars templates will be in their own folder.	DONE
Coding Standards	Handlebars layouts will be in their own folder.	DONE
Coding Standards	Images will be in their own folder.	DONE
Coding Standards	The main branch for frontend code will be the 'frontend-development' branch.	DONE
Coding Standards	The main branch for backend code will be the 'backend-development' branch.	DONE
Coding Standards	Code that is ready to be tested will be moved from the 'development' branch to the 'testing' branch for the frontend code.	ISSUE, no testing branch
Coding Standards	Code that is ready to be tested will be moved from the 'development' branch to the 'testing' branch for the backend code.	ISSUE, no testing branch
Coding Standards	Code will only be pushed after pulling first.	DONE

Availability	The website will not restrict users based on their geographic location.	DONE
Availability	The website's URL shall be reliable to access its contents.	DONE
Availability	The website's URLs shall be simple and short.	ON TRACK, some URL's use query parameters which increase their length
Availability	The website will load in at least 20 seconds unless there is a system error.	DONE
Scalability	The website will be designed such that it can be migrated to higher-performance hardware if demand increases.	DONE
Fault Intolerance	A backup of the virtual server will be performed daily.	DONE
Fault Intolerance	The virtual server will be restored from a backup in the event of a total system failure.	DONE
Expected Load	The website is expected to handle 10-20 users at a time.	ON TRACK
Privacy	The website shall not save the users CVV number during the checkout process.	ON TRACK, checkout process is priority 2
Privacy	The website shall not save the users CVV number after the checkout process.	ON TRACK, checkout process is priority 2
Privacy	This website shall not sell user personal information to third parties.	DONE
Privacy	The website will not share private user information with other users unless where explicitly done by	DONE

	the user.	
Privacy	The website shall not store any user medical information.	DONE
Privacy	The website will not store users' IP addresses.	DONE
Privacy	The website will not share users' IP addresses.	DONE
Legal	The website will not purposely publish any false or fraudulent information in any form.	DONE
Legal	The website shall have Terms and Conditions which users must consent to before logging in.	DONE
Legal	The website shall notify the user when the Terms and Conditions have been changed.	ISSUE, we would have to develop a separate process to send emails out
Legal	The website shall not let the user continue until they have accepted the updated Terms and Conditions.	DONE, there is a message on the Registration Page explaining this agreement
Look and Feel	The website will use a uniform color palette on all pages of the site.	ON TRACK
Look and Feel	The website will have a responsive design.	ON TRACK
Look and Feel	The website will scale to the width of the screen.	ON TRACK
Look and Feel	The website will scale to the height of the screen.	ON TRACK
Look and Feel	The website will be optimized for tablet clients.	ISSUE, we

		too have many style files
Look and Feel	The website will be optimized for desktop clients.	ON TRACK
Look and Feel	The website will be optimized for mobile clients.	ISSUE, same as above
Look and Feel	The website will have readable text font size.	ON TRACK
Look and Feel	The website will have uniform font styles.	ON TRACK
Content	The website will have no typos.	ON TRACK
Content	The website will have alt tags for images.	ON TRACK
Look and Feel	The website will accurately display currency to two decimal places.	DONE

## 7. Contributions

### Taylor:

- DB procedures, Server code for request handling and DB interaction, Route Planner client and server
- Testing Plan for Search for A Flight

### Yi Liu:

- Testing plan for Learn About the Team members

### Luis Alfaro:

- Made updates to the database procedures and model, fixed DB errors
- Integrated DB procedures into server code and request handlers

### Robin Fernando:

### Ruja Rajbhandari:

- Configured the navbar
- Testing Plan for Reopen a Saved Trip

### Kajeme Cheneque:

- Performed testing on the webpages
- Testing Plan for View A Previous Trip
- Fixed broken links on Registration and Reset Password

### Jesus Correa:

- Configured request handlers for About Us, Individual about pages, Terms and Conditions, Privacy page
- Designed Home Page and Photo Gallery, About Us, Individual about pages, Terms and Conditions, Privacy page
- Testing Plan for Upload an Image