### API Documentation for Trip Tracker Application

#### 1. ****Login API****

* **Endpoint**: /log
* **Method**: GET, POST
* **Description**: This endpoint allows users to log in to the application using their email and password.

##### Request:

* **GET**: Renders the login page.
* **POST**: Accepts the following parameters in the request body:
  + email: User's email address (required).
  + password: User's password (required).

##### Response:

* **200 OK**: If login is successful, the user is redirected to the home page and a success message is flashed.
* **400 Bad Request**: If the credentials are incorrect or the email does not exist.

#### 2. ****Logout API****

* **Endpoint**: /logout
* **Method**: GET
* **Description**: This endpoint logs out the current user and redirects to the login page.

##### Request:

* **GET**: Logs out the current user.

##### Response:

* **200 OK**: Logs out successfully and redirects to the login page.

#### 3. ****Sign Up API****

* **Endpoint**: /sign\_up
* **Method**: GET, POST
* **Description**: This endpoint allows new users to sign up for an account.

##### Request:

* **GET**: Renders the sign-up page.
* **POST**: Accepts the following parameters in the request body:
  + email: User's email address (required).
  + firstName: User's first name (required).
  + password1: User's password (required).
  + password2: User's confirmation of the password (required).

##### Response:

* **200 OK**: If the account is created successfully, the user is logged in and redirected to the home page with a success message.
* **400 Bad Request**: If the email already exists, passwords don't match, or other validation errors.

#### 4. ****Add Trip API****

* **Endpoint**: /addTrip
* **Method**: POST
* **Description**: This endpoint allows logged-in users to add a new trip to their profile.

##### Request:

* **POST**: Accepts the following parameters in the request body (form data):
  + place: Name of the destination (required).
  + days: Number of days for the trip (required).
  + month: Month of the trip (required).
  + budget: Estimated budget for the trip (optional).
  + accommodation: Accommodation details (optional).
  + transport: Transportation details (optional).
  + local\_travel: Local travel details (optional).
  + places\_visited: Places visited during the trip (optional).
  + food\_recommendations: Food recommendations (optional).
  + trip\_description: A description of the trip (optional).

##### Response:

* **200 OK**: The trip is added successfully, and the user is redirected to their trip list page with a success message.
* **400 Bad Request**: If any required field is missing.

#### 5. ****View My Trips API****

* **Endpoint**: /mytips
* **Method**: GET
* **Description**: This endpoint renders a page showing all trips added by the currently logged-in user.

##### Request:

* **GET**: Fetches the trips for the logged-in user.

##### Response:

* **200 OK**: Renders the mytips.html page displaying the user's trips.

#### 6. ****Search Trips API****

* **Endpoint**: /search
* **Method**: GET
* **Description**: This endpoint allows users to search for trips by place name.

##### Request:

* **GET**: Accepts a query parameter in the URL:
  + query: The search term to filter trips based on place name.

##### Response:

* **200 OK**: Returns a list of trips that match the search term, along with their details:
  + place: Destination place name.
  + days: Number of days for the trip.
  + trip\_description: Description of the trip.
  + budget: Estimated budget for the trip.
  + date\_added: Date the trip was added.
  + user\_name: Name of the user who added the trip.

##### Example Input:

GET /search?query=Paris

##### Example Output:

{

"destinations": [

{

"place": "Paris",

"days": "7",

"trip\_description": "A cultural trip to Paris.",

"budget": "2000",

"date\_added": "2024-11-10",

"user\_name": "John"

}

]

}

#### 7. ****Delete Trip API****

* **Endpoint**: /delete\_trip/<int:trip\_id>
* **Method**: DELETE
* **Description**: This endpoint allows users to delete a specific trip they have added.

##### Request:

* **DELETE**: The trip\_id parameter is included in the URL to specify which trip to delete.

##### Response:

* **200 OK**: Trip deleted successfully.
* **404 Not Found**: If the trip does not exist or if the user is not the owner of the trip.

#### 8. ****Get User's Trips API****

* **Endpoint**: /get\_user\_trips
* **Method**: GET
* **Description**: This endpoint returns a list of all trips added by the current logged-in user.

##### Request:

* **GET**: No parameters required.

##### Response:

* **200 OK**: A list of the user's trips in JSON format.

##### Example Output:

[

{

"id": 1,

"place": "Paris",

"date": "2024-11-10",

"days": "7",

"budget": "2000",

"accommodation": "Hotel",

"transport": "Airplane",

"local\_travel": "Subway",

"places\_visited": "Eiffel Tower, Louvre",

"food\_recommendations": "Croissants, Baguette",

"trip\_description": "A cultural trip to Paris."

}

]

#### 9. ****Generate Itinerary API****

* **Endpoint**: /submit
* **Method**: POST
* **Description**: This endpoint generates an itinerary for a trip based on user inputs (place, visit date, and requirements).

##### Request:

* **POST**: Accepts the following parameters in the request body (form data):
  + place: Name of the destination (required).
  + visit\_date: The date of the visit (required).
  + days: Number of days for the trip (required).
  + requirements: Any special requirements or preferences for the trip (optional).

##### Response:

* **200 OK**: Returns the generated itinerary formatted in HTML.

##### Example Output:

<div class="day">

<h2>Day 1</h2>

<p>Visit the Eiffel Tower in the morning. Enjoy a Baguette for lunch. Explore Montmartre in the afternoon.</p>

</div>

#### 10. ****Generate Day Plan API****

* **Endpoint**: /generate\_day\_plan
* **Method**: POST
* **Description**: This endpoint generates a detailed day plan for the user's selected location, including time-based activities.

##### Request:

* **POST**: Accepts the location in the session, fetched via the /set\_location endpoint.

##### Response:

* **200 OK**: Returns the formatted day plan.

##### Example Output:

{

"day\_plan": "<b>1.</b> Visit the Eiffel Tower in the morning. <b>2.</b> Lunch at a local café."

}