**TechFest-Hospitality Documentation**

**Overview**

**TechFest-Hospitality** is a web application developed during an internship at Techfest. It is designed to streamline accommodation bookings for Techfest adventures. The platform allows users to upload CSV files for group and hostel allocations, processes the data, and provides an option to download the allocation results.

**Features**

* **CSV File Upload**: Users can upload two CSV files—one for group allocations and one for hostel allocations.
* **Automatic Processing**: The application processes the uploaded CSV files to generate accommodation allocations.
* **Download Results**: Users can download the processed results as a CSV file.

**Usage**

**File Upload**

1. **Group CSV File**:
   * Click on the "Upload Group CSV file" button.
   * Select the CSV file containing group allocation data from your local machine.
2. **Hostel CSV File**:
   * Click on the "Upload Hostel CSV file" button.
   * Select the CSV file containing hostel allocation data from your local machine.

**Submit and Download**

1. **Submit Files**:
   * Click on the "Submit Both CSVs and Download" button.
   * The application will process the uploaded files and generate the allocation results.
2. **Download Results**:
   * Upon successful processing, the application will automatically trigger a download of the allocation results as a CSV file named allocations.csv.

**Logic**

**Handling File Uploads**

* The handleFile1Change and handleFile2Change functions handle the file selection for group and hostel CSV files, respectively. These functions update the state with the selected files and their names.

**Form Submission**

* The onSubmit function handles the form submission:
  + It checks if both files are selected.
  + If files are missing, it displays an error message.
  + It creates a FormData object and appends the selected files.
  + It sends a POST request to the server with the form data for processing.

**File Processing and Download**

* The server processes the uploaded files and generates the allocation results.
* Upon successful processing, the downloadFile function is called to download the results as a CSV file.
* If the processing fails, an error message is displayed.

**Loading Indicator**

* A loading animation is displayed while the files are being processed and downloaded, providing visual feedback to the user.

**Error Handling**

* Appropriate error messages are displayed for file upload issues, submission failures, and download errors.
* Messages are cleared after a short duration to keep the interface clean.

**Edge Cases**

* **File Not Selected**: The application checks if both files are selected before submission and prompts the user if any file is missing.
* **Server Errors**: If the server fails to process the files or download the results, appropriate error messages are displayed.
* **Invalid File Types**: The application should handle cases where non-CSV files are uploaded, though this is not explicitly covered in the current implementation.

**Future Enhancements**

* **Live Deployment**: Move the application from localhost to a live server.
* **Enhanced Error Handling**: Improve error messages and validation for file types and content.
* **User Authentication**: Add user authentication to secure file uploads and downloads.
* **Detailed Logging**: Implement detailed logging for better debugging and monitoring.