

# Group #8: Travel Assist

Zavier, Jasper, and Joon

## **Description:**

- Problem: Group trips are notoriously hard to plan and coordinate. Keeping track of all the tasks that need to be completed can be incredibly challenging to do across instant messaging.
- Solution: The travel coordinator aids in planning both solo and group trips by providing a platform to keep all your information in one place: dates, locations, budgets, and itineraries can all be viewed and adjusted here.

## **Key Features:**

- Task List: a to-do-list-like feature to assist individuals in planning what tasks need to be done before, during, or after the trip. Optionally adding a modifiable packing list.
- Messaging between group members.
- Budgeting features: Group members can include fees for travel, food, fun, etc.
- Itinerary: Group members can add events/activities they are interested in or intend on participating in, including when, where, what it is, and optional costs.

[Link to tagged repo](#)

[Link to issues/milestones](#)

# Meet ✨*The Team*✨

## Jasper

- Role: homepage manager
- 3 Assigned Issues:
  - Main Page UI
  - Abstract class implementation
  - New trip implementation

## Zavier

- Role: messaging manager
- 3 assigned issues:
  - Trips Data
  - Messaging UI
  - Messaging Data

## Joon

- Trip-page manager
- 3 Assigned Issues:
  - Trip-page UI
  - Sidebar Connection
  - Todo UI

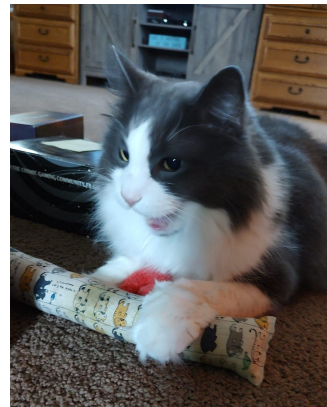
# Zavier Morales

Email: [zmorales@umass.edu](mailto:zmorales@umass.edu)

**About me:** I'm a Junior in CS, and I'm interested in biotechnology, machine learning, and LOTR! I have some experience in TS/JS, React, Vue, Html, CSS, and Google APIs, but I learn fast and I'm willing to work with any stack we choose.

**Favorite quote:** "Some infinities are bigger than other infinities." — John Green, *The Fault in Our Stars*

**Reflection:** Coordination for trips has always been a headache for friends and I, especially agreeing on times, tickets, pricing, and even finding a time to meet up to talk about the trip. This app seems like a really good idea to centralize all of these problems, and also it includes a range of tools to implement, which sounds fun.



*Jasper McCormack (He/Him)*

*caleighmccor@umass.edu*

*I'm a transfer student from Holyoke Community College and in my Junior year. I've taken CS 220, have some HTML experience, and have worked with GUI applications in Java already. I have an interest in visual design, digital or physical artwork, and website development.*

*I believe that a travel planner is a fun and interesting application to build since I personally have trouble organizing my personal plans and budgeting for trips, and coordinating with other people to complete tasks can be incredibly difficult with generic messaging and calendar apps. A consolidated application would ease a lot of stress.*



*I don't know anything with any certainty, but the sight of the stars makes me dream"*  
*- Vincent Van Gogh*



# Assigned Work Summary - Joon

In this milestone, I focused on developing the trip management JavaScript functionality. I worked on connecting the multi-view sidebar, implementing local data storage for trip information, and creating a todo list with add/delete functionality. My CSS still needs refinement, but I prioritized making the database connections work properly.

Pull- request: <https://github.com/BakingPancakes/Travel-Assistant/pull/38>

Issues: <https://github.com/BakingPancakes/Travel-Assistant/issues/6>

Future Issues: <https://github.com/BakingPancakes/Travel-Assistant/issues/39>

<https://github.com/BakingPancakes/Travel-Assistant/issues/40>

# Feature Demonstration - Joon

TO-DOs

BEFORE

Write todo

Add

BOOKING

delete

HW

delete

DURING

Write todo

Add

SSS

delete

AFTER

Write todo

Add

1222

delete

PROFILE

Home

Your Trips

Messages

Calendar

This is HOME page

show the homepage.

"TRIP NAME"

🔔

Edit Trip

Location

home

Traveler

me

Traveler's Companion

idk

From

2025. 03. 31. 📅

To

2025. 04. 18. 📅

Submit Trip

I implemented three key features: a multi-view interface that switches pages when sidebar items are clicked, a main page that saves basic information to local storage (preparing for future server integration), and a todo list on the sidebar with add and delete capabilities to help users organize their tasks.

# Code Explanation - Joon

```
function saveTripData() {  
  const travelerData = {  
    location: document.getElementById('location').value || '',  
    traveler: document.getElementById('traveler').value || '',  
    companions: document.getElementById('companions').value || '',  
    from: document.getElementById('from').value || '',  
    to: document.getElementById('to').value || ''  
  };  
  
  localStorage.setItem('tripData', JSON.stringify(travelerData));  
  displayTripData();  
}  
  
function loadTripData() {  
  const savedData = localStorage.getItem('tripData');  
  
  if (savedData) {  
    const travelerData = JSON.parse(savedData);  
  
    document.getElementById('location').value = travelerData.location || '';  
    document.getElementById('traveler').value = travelerData.traveler || '';  
    document.getElementById('companions').value = travelerData.companions || '';  
    document.getElementById('from').value = travelerData.from || '';  
    document.getElementById('to').value = travelerData.to || '';  
  
    displayTripData();  
  }  
}
```

```
function saveTripData() {  
  const travelerData = {  
    location: document.getElementById('location').value || '',  
    traveler: document.getElementById('traveler').value || '',  
    companions: document.getElementById('companions').value || '',  
    from: document.getElementById('from').value || '',  
    to: document.getElementById('to').value || ''  
  };  
  
  localStorage.setItem('tripData', JSON.stringify(travelerData));  
  displayTripData();  
}  
  
function loadTripData() {  
  const savedData = localStorage.getItem('tripData');  
  
  if (savedData) {  
    const travelerData = JSON.parse(savedData);  
  
    document.getElementById('location').value = travelerData.location || '';  
    document.getElementById('traveler').value = travelerData.traveler || '';  
    document.getElementById('companions').value = travelerData.companions || '';  
    document.getElementById('from').value = travelerData.from || '';  
    document.getElementById('to').value = travelerData.to || '';  
  
    displayTripData();  
  }  
}
```

```
const todoValue = todoInput.value;  
  
const li = document.createElement("li");  
li.classList.add("todo-item");  
  
const todoText = document.createElement("span");  
todoText.textContent = todoValue;  
  
const deleteBtn = document.createElement("button");  
deleteBtn.textContent = "delete";  
deleteBtn.classList.add("delete-btn");  
deleteBtn.onclick = () => {  
  li.remove();  
}  
  
li.appendChild(todoText);  
li.appendChild(deleteBtn);  
  
todoList.appendChild(li);  
  
todoInput.value = "";
```

The images show three key components of my implementation: the HTML structure for the multi-view interface and the JavaScript functions for data management. The code organizes different views that can be toggled, while the JavaScript functions handle saving and retrieving trip data using localStorage for persistence between sessions.

The todo code creates a simple task management system. It captures user input to generate list items containing both task text and delete buttons, adds them to the todo list, and clears the input field afterward. Each delete button removes its parent item when clicked, allowing users to easily add and remove tasks as needed.



# Challenges and Insights- Joon

I gained better understanding of multi-view implementation but faced some difficulties. Our team is still working on consolidating our files into a single HTML structure. I implemented data storage and retrieval functionality but encountered an issue where data appears to be lost after editing and saving, though it reappears after refreshing the page.

# Future Improvements & Next Steps - Joon

I need to refine the data persistence functionality to ensure data isn't temporarily lost during edit operations. The CSS needs improvement for better visual presentation. I also plan to enhance the todo list functionality and improve integration between the different views to create a more seamless user experience.

I just had conflict with merge my branch to main branch I uploaded with edit files on main branch but I have to figure out how to branch direct to my branch.

# Timeline - Joon

April 9th:

- I added todo list functionality to our project! Now users can add and delete items from the todo list using simple JavaScript.

April 10th:

- I implemented local storage for traveler information. When users enter their details, the information is saved in the browser's local database so it's still there when they come back later.

April 11th:

- I connected the sidebar navigation to create a multi-view interface! Now users can click different sidebar options to switch between different pages without reloading the whole website.

# Assigned Work Summary - Xavier

- Dynamically adding messages
  - Re-formatting message bubbles to allow messages to be flexibly added to the view
- Multi-view UI: view changes according to which person you click on

## Timeline - Xavier

# Feature Demonstration - Xavier



# Code Explanation - Xavier

## Challenges/Insights - Xavier

# Future Improvements/Next Steps - Xavier

-

# Assigned Work Summary - Jasper McCormack

- Completion of homepage mockup
- Assigned Outline of EventBus, services, and Abstract classes
  - I volunteered to begin a majority of the abstract class implementation such as BaseComponent.js, Service.js, etc. because I had a bit more time to work on the project.
- Creation of New Trips
  - Still working on implementation for creating trips to be viewed on trip pages and list views.
- Pull Requests:
  - <https://github.com/BakingPancakes/Travel-Assistant/pull/27>
  - <https://github.com/BakingPancakes/Travel-Assistant/pull/33>
  - <https://github.com/BakingPancakes/Travel-Assistant/pull/41>
- Closed Issues:
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/5>

# Assigned Work - Jasper McCormack

- Outstanding issues:
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/28> (Creating New Trip)
    - While I have made significant headway in this feature, it still is incomplete
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/37> (TripInputComponent)
    - JavaScript file is completed for now, but css styling still needs to be added
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/35> (SidebarComponent)
    - While sidebar JavaScript and css are written, I don't want to close the issue until I have had a better chance to test it.
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/36> (ScreenComponent)
    - Still working out how to input multi-view.
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/29> (Empty Input Validation)
    - Given this will need to be reused, I need to plan out the best implementation
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/30> (Date Input Validation)
    - Same reason as empty input validation. Want to implement for optimal reusability.

# Timeline - Jasper

April 9th:

- Overcame the roadblock of table mockups from milestone #4, and rewrote the css styling... a third time.

April 10th:

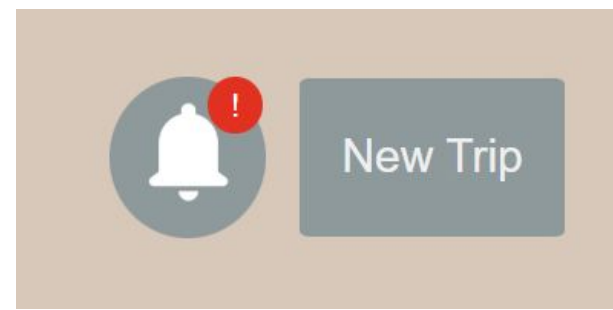
- Further refinement of the homepage mockup, ensuring class names all worked correctly and added in nicer colors. Merged the work into main
- Began working on implementation aspects such as components, js files.

April 11th:

- Created abstract classes BaseComponent and Services, began to implement other needed classes for trip creation feature, such as EventHub, TripRepository JavaScript files, as well as base64 and fetch.js from the task example we were given.
- Creation of the package for the application, as well as a primary html file. Also added a test html page.



# Feature Demonstration - Jasper



Feature: Creation of new Trips

Currently in progress. The goal of the feature is to implement a form opened by the 'new trip' button that at the very least permits a user to add in basic information about their trip, submit it, and load the information about the trip onto the specified page, or dynamically load in an entry into the trip list on the homepage.

Includes:

- A. Dynamic content updates, since it would include displaying new items in the trip list
- B. Form validation and user feedback, since a user would be required to fill out the necessary fields for a trip
- C. Event handling, if they elected to switch to the dedicated trip page
- D. Multi-view user interface, since the components of each page will be loaded dynamically.
- E. IndexedDB to temporarily save the trip through reloads of the page.

I would estimate on completion that this would warrant between 3 and 4 points.

Due to a lack of proper planning on my end, the feature isn't entirely completed. However, the component to retrieve user input is coded and will be implemented shortly. Repositories (local, fake, and remote) have been implemented to save the form input. In addition, Dynamic content updates are being implemented as components are completed. I have been working on different sections of the feature at different times.

# Code Explanation - Jasper

The code to the right is the implementation of the sidebar component that can be reused across different pages. In addition, the code can be modified to fit other components, not just the sidebar.

The greatest challenge so far is designing the modularity of the components so that different components can be paired together. Still working on a solution, but I think that I can modify the appcontroller in the task application to work for the 'screen' in our application.

```
import { BaseComponent } from '../BaseComponent/BaseComponent';

export class SidebarComponent extends BaseComponent {
  #container = null;

  constructor() {
    super();
    this.loadCSS('SidebarComponent');
  }

  render() {
    if (this.#container) {
      return this.#container;
    }

    this.#createContainer();
    this.#setupContainerContent();
    this.#attachEventListeners();

    return this.#container;
  }

  // Creates the container element for the component
  #createContainer() {
    this.#container = document.createElement('div');
    this.#container.classList.add('screen_sidebar');
  }

  #setupContainerContent() {
    this.#container.innerHTML = `
    <div id="profile-div">
      <button id="button-profile"></button>
    </div>
    <button id="button-home" class="sidebar_button">Home</button>
    <button id="button-trips" class="sidebar_button">Trips</button>
    <button id="button-messages" class="sidebar_button">Messages</button>
    <button id="button-calendar" class="sidebar_button">Calendar</button>
  `;
  }

  // Attach the event listeners to the component
  #attachEventListeners() {
    const profileBtn = this.#container.querySelector('#button-profile');
    const homeBtn = this.#container.querySelector('#button-home');
    const tripBtn = this.#container.querySelector('#button-trips');
    const messageBtn = this.#container.querySelector('#button-messages');
    const calendarBtn = this.#container.querySelector('#button-calendar');

    profileBtn.addEventListener("click", () =>
      this.#handleProfileClick()
    );

    homeBtn.addEventListener("click", () =>
      this.#handleHomeSwitch()
    );

    tripBtn.addEventListener("click", () =>
      this.#handleTripSwitch()
    );

    messageBtn.addEventListener("click", () =>
      this.#handleMsgSwitch()
    );

    calendarBtn.addEventListener("click", () =>
      this.#handleCalendarSwitch()
    );
  }
}
```



Welcome

Home

Trips

Messages

Calendar

Trip

some trip

To-Do's

Task

some task

Tasks Remain

# Challenges/Insights - Jasper

Converting all components from the html mockup into JavaScript has proven to be more tedious than I anticipated. Because we want to be able to reuse certain components like the sidebar, determining how to implement certain aspects of how the screen will render has been difficult.

Yet again, the balance of coursework has gotten to me, but I have done significantly better than the last time. However, because of my nitpicking about aesthetic details, I didn't complete as much in terms of features as I should have. My intention is to step away from modification of new aesthetics until I can properly implement my desired functionality.

In other, better news, I was able to discuss my situation with my employer and reduce my hours in order to have more time for assignments much like this one.

# Future Improvements/Next Steps - Jasper

- User Profiles
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/12>
- Login/Logout features
  - <https://github.com/BakingPancakes/Travel-Assistant/issues/24>