Story: Driv- 6
Sub Task: Driv- 65
Determine relevant search fields to filter discussions.

Story: Driv- 6
Sub Task: Driv- 63
Develop backend interface to display all active discussions

Story: Driv- 7
Sub Task: Driv-68
Determine the fields for posts

Story: Driv- 7
Sub Task: Driv-66
Develop frontend button and form for adding posts

Story: Driv- 6
Sub Task: Driv- 62
Develop frontend list interface with search bar and filters.

Story: Driv- 6
Sub Task: Driv- 64
Develop logic to allow users to filter through discussion posts

Story: Driv- 9
Sub Task: Driv-69
Develop frontend template to leave comments

Story: Driv- 7
Sub Task: Driv-67
Develop backend logic to store newly created posts

Story: Driv- 10
Sub Task: Driv-71
Integrate a calendar component into the React project to display events.

Story: Driv- 10
Sub Task: Driv-72
Create a form to input event details and integrate it with the calendar.

Story: Driv- 12
Sub Task: Driv-75
Enhance calendar component to display events

Story: Driv- 9
Sub Task: Driv-70
Develop backend logic to store and display comments

Story: Driv- 10
Sub Task: Driv-73
Connect the event form to the calendar so that events added through the form appear on the calendar.

Story: Driv- 10
Sub Task: Driv-74
Create a backend API to save, retrieve, and manage events in the MongoDB database.

Story: Driv- 12
Sub Task: Driv-77
Allow students to RSVP for events through notification, usch as accepting or rejecting invite

Story: Driv- 12
Sub Task: Driv-76
Develop modal to display detailed event information

Story: Driv- 79
Sub Task: Driv-80
Develop API endpoint to remove an attendee.

Story: Driv- 12
Sub Task: Driv-78 Test the event viewing and RSVP

Story: Driv- 30
Sub Task: Driv-58
Design and develop future grades input.

Story: Driv- 79
Sub Task: Driv-81
Develop API endpoint to remove an attendee.

Story: Driv- 13
Sub Task: Driv-57
Connect frontend to backend

Story: Driv- 79
Sub Task: Driv-82 Add frontend functionality to remove users from an event

Story: Driv- 30
Sub Task: Driv-59 Integrate future grades calculation with current GPA

Story: Driv- 13
Sub Task: Driv-55
Design and develop input form

Legend

B depends on A

User Story Driv-6: Searching and Filtering Discussions

SubTask Driv-65: Determine relevant search fields

- **Reason**: This is the first step as it is need to know what fields will be searched to provide meaningful search results.

SubTask Driv-62: Develop frontend list interface with search bar and filters (depends on Driv-65)

- **Depends on**: Driv-65
- **Reason**: The UI design and functionality depend on knowing what the search fields are.

SubTask Driv-63: Develop backend interface to display all active discussions (depends on Driv-65)

- **Depends on**: Driv-65
- **Reason**: The backend needs to know what fields will be included in the search.

SubTask Driv-64: Develop logic to allow users to filter through discussion posts (depends on Driv-62 and Driv-63)

- **Depends on**: Driv-62, Driv-63
- **Reason**: Filtering logic requires both the frontend interface to input the filters and the backend logic to apply these filters to the discussion data.

User Story Driv-7: Creating a New Discussion Post

SubTask Driv-68: Determine the fields for posts

- **Reason**: This is the first step as it is need to know what information will be included in a new discussion post.

SubTask Driv-66: Develop frontend button and form for adding posts (depends on Driv-68)

- **Depends on**: Driv-68
- **Reason**: The frontend form needs to include all the fields that have been determined for a new discussion post.

SubTask Driv-67: Develop backend logic to store newly created posts (depends on Driv-68)

- **Depends on**: Driv-68
- **Reason**: The backend logic must know what fields it needs to store in the database when a new post is created.

User Story Driv-9: Commenting on Discussion Posts

SubTask Driv-69: Develop frontend template to leave comments

- **Reason**: This is the first step in allowing users to comment on posts by providing a UI for inputting comments.

SubTask Driv-70: Develop backend logic to store and display comments (depends on Driv-69)

- **Depends on**: Driv-69
- **Reason**: The backend logic must know how the comment data is structured and how to store it. This requires the frontend template to be defined first.

User Story Driv-10: Adding Events to the Calendar

SubTask Driv-71: Integrate calendar component into React project

- **Reason**: This is the foundational step for adding events, as it provides the component where events will be displayed.

SubTask Driv-72: Create a form to input event details (depends on Driv-71)

- **Depends on**: Driv-71
- **Reason**: The form for adding events needs to be integrated with the calendar component so that events can be displayed once added.

SubTask Driv-73: Connect event form to the calendar (depends on Driv-72)

- **Depends on**: Driv-72
- **Reason**: The form needs to be functional and able to add events to the calendar, requiring the connection between the form and the calendar.

SubTask Driv-74: Create backend API to manage events (depends on Driv-73)

- **Depends on**: Driv-73
- **Reason**: The backend needs to store the event data that the form collects and the calendar displays, so it needs to know how the form and calendar interact.

User Story Driv-12: Viewing and RSVP for Events

SubTask Driv-75: Enhance calendar component to display events

- **Reason**: This is the foundational step to view events in the calendar.

SubTask Driv-76: Develop modal to display detailed event information (depends on Driv-75)

- **Depends on**: Driv-75
- **Reason**: The modal for detailed information must be linked to the calendar component that displays the events.

SubTask Driv-77: Allow students to RSVP for events (depends on Driv-76)

- **Depends on**: Driv-76
- **Reason**: The RSVP accept or reject functionality requires information from event to be displayed and it to show in notifications component

SubTask Driv-78: Test the event viewing and RSVP feature (depends on Driv-77)

- **Depends on**: Driv-77
- **Reason**: Testing ensures that all components (viewing events, detailed information, RSVP) work together seamlessly.

User Story Driv-79: Removing Self from Event

SubTask Driv-80: Develop API endpoint to remove an attendee

- **Reason**: This is the foundational step to allow users to remove themselves from an event.

SubTask Driv-81: Ensure the Event model supports removal (depends on Driv-80)

- **Depends on**: Driv-80
- **Reason**: The Event model must be capable of handling attendees removaing themselves, which requires the API endpoint to be defined first.

SubTask Driv-82: Add frontend functionality to remove users from an event (depends on Driv-81)

- **Depends on**: Driv-81
- **Reason**: The frontend needs to interface with the backend to allow users to remove themselves, requiring the backend support to be in place.

User Story Driv-30: Calculating Potential GPA

SubTask Driv-58: Design and develop future grades input form

- **Reason**: This is the foundational step for calculating potential GPA.

SubTask Driv-59: Integrate future grades calculation with current GPA (depends on Driv-58)

- **Depends on**: Driv-58
- **Reason**: The calculation logic needs the input from the form to perform the calculation.

User Story Driv-13: Inputting Grades into GPA Calculator

SubTask Driv-55: Design and develop input form

- **Reason**: This is the foundational step for inputting grades.

SubTask Driv-57: Connect frontend to backend (depends on Driv-55)

- **Depends on**: Driv-55
- **Reason**: The backend must store the grades entered through the frontend form, requiring the form to be defined first.

Critical Path:

Driv-65: Determine relevant search fields

Driv-62: Develop frontend list interface with search bar and filters (depends on Driv-65)

Driv-63: Develop backend interface to display all active discussions (depends on Driv-65)

Driv-64: Develop logic to allow users to filter through discussion posts (depends on Driv-62 and Driv-63)

Driv-68: Determine the fields for posts

Driv-66: Develop frontend button and form for adding posts (depends on Driv-68)

Driv-67: Develop backend logic to store newly created posts (depends on Driv-68)

Driv-69: Develop frontend template to leave comments

Driv-70: Develop backend logic to store and display comments (depends on Driv-69)

Driv-71: Integrate calendar component into React project

Driv-72: Create a form to input event details (depends on Driv-71)

Driv-73: Connect event form to the calendar (depends on Driv-72)

Driv-74: Create backend API to manage events (depends on Driv-73)

Driv-75: Enhance calendar component to display events

Driv-76: Develop modal to display detailed event information (depends on Driv-75)

Driv-77: Allow students to RSVP for events (depends on Driv-76)

Driv-78: Test the event viewing and RSVP feature (depends on Driv-77)

Driv-80: Develop API endpoint to remove an attendee

Driv-81: Ensure the Event model supports removal (depends on Driv-80)

Driv-82: Add frontend functionality to remove users from an event (depends on Driv-81)

Driv-58: Design and develop future grades input form

Driv-59: Integrate future grades calculation with current GPA (depends on Driv-58)

Driv-55: Design and develop input form

Driv-57: Connect frontend to backend (depends on Driv-55)

Explain what you do to keep your sprint in schedule

We have ensured that the tasks that have other stories/subtasks depending on it, to be prioritize first and identifying these as these tasks affect the timeline for our Sprint 2, so when we focus on them we ensure we spend our time on those to minimize delays.

We also used frequent standups during our development cycle to help us review our progress and our blockers and to adjust story point priorities as we needed when we realized other tasks that are in the critical path. These meetings helped us focus on key updates to each other.

We also ensured during the initial planning of the user stories and sprint 2 in general that we broke down the user story into manageable tasks to work up to the completion. This helped with our initial estimation of the user story points and also gave us a good idea of the task dependencies.

We also tried to identify risks and then the following contingency plans.This lead us to effectively communicating, involving maintaining open communication of the development and encourage the team when they are stuck.We used instagram mostly.