Story: Driv- 16
Sub Task: Driv-83 Determine layout and features

Story: Driv- 16
Sub Task: Driv-84 Develop frontend user interface.

Story: Driv- 16
Sub Task: Driv-85 Develop logic to help users search through tags

Story: Driv- 16
Sub Task: Driv-85 Develop backend logic to help users post.

Story: Driv- 17
Sub Task: Driv-87 Design Course Input Form.

Story: Driv- 17
Sub Task: Driv-88 Develop Course Addition Functionality

Story: Driv- 17
Sub Task: Driv-89 Develop Course Editing and Deletion Features from an event

Story: Driv- 18
Sub Task: Driv-90 Define Task Schema and Create API Endpoints

Story: Driv- 18
Sub Task: Driv-91 Develop Frontend Components for Task Management

Story: Driv- 18
Sub Task: Driv-92 Implement Task Priority Sorting and Filtering

Legend

B depends on A

User Story Driv-16: Viewing and Adding Links to Resource Library

- **SubTask Driv-83: Determine layout and features within resource library page**
  - **Reason:** This is the first step.
- **SubTask Driv-84: Develop frontend user interface for resource library**
  - **Depends on:** Driv-83
  - **Reason:** The design depends on Driv-83.
- **SubTask Driv-85: Develop logic to help users search through resources based on tags**
  - **Depends on:** Driv-84
  - **Reason:** The search logic needs to integrate and search.
- **SubTask Driv-86: Develop backend logic to help users post and retrieve resources**
  - **Depends on:** Driv-84
  - **Reason:** The backend logic must align with the UI to handle data input and retrieval efficiently.

User Story Driv-17: Adding Courses to Degree Planner

- **SubTask Driv-87: Design Course Input Form**
  - **Reason:** This is the foundational step as it defines the structure and fields required for adding courses.
- **SubTask Driv-88: Develop Course Addition Functionality**
  - **Depends on:** Driv-87
  - **Reason:** The addition functionality requires the input form design to collect and process course data.
- **SubTask Driv-89: Develop Course Editing and Deletion Features**
  - **Depends on:** Driv-88
  - **Reason:** Editing and deletion functionalities need the basic course addition logic to be implemented first.

User Story Driv-18: Task Management in Degree Planner

- **SubTask Driv-90: Define Task Schema and Create API Endpoints**
  - **Reason:** This is the defining step for toDo management, defining the structure and endpoints required for managing toDos.
- **SubTask Driv-91: Develop Frontend Components for Task Management**
  - **Depends on:** Driv-90
  - **Reason:** The frontend components need the API to fetch, display, and manage toDos.
- **SubTask Driv-92: Implement Task Priority Sorting and Filtering**
  - **Depends on:** Driv-91
  - **Reason:** Sorting and filtering require the frontend components to be in place to apply these functionalities.

Critical Path:

Driv-84: Develop frontend user interface for resource library

Driv-85: Develop logic to help users search through resources based on tags (depends on Driv-84)

Driv-87: Design Course Input Form

Driv-88: Develop Course Addition Functionality (depends on Driv-87)

Driv-89: Develop Course Editing and Deletion Features (depends on Driv-88)

Driv-90: Define Task Schema and Create API Endpoints

Driv-91: Develop Frontend Components for Task Management (depends on Driv-90)

Driv-92: Implement Task Priority Sorting and Filtering (depends on Driv-91)

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.