Group Members: Connor O'Neill, Jace Sullivan, Jason

Overview:

The project that we intend to create is a Golf Course Agronomy Department Management System. This is designed to aid in the management of the various tasks that an agronomy crew needs to do at a golf course. This includes but is not limited to which surfaces have been mowed and when(greens, tees, approaches, rough, fairways), when the last time chemicals were distributed on the course and which ones were distributed, any areas with special needs, and create a job board for the day.

In addition, it can create a budget, track and categorize expenses, visualize data, and view financial summaries of the budget on the golf course. It can also keep track of equipment and show how much there is, if there is a shortage, and if some of it is being maintained. The crew will be able to view the daily tasks ahead and allow for each person to know what is going on at the course.

Features:

Allows for the creation of accounts with no password

Tracks maintenance that has occurred

Can create a schedule for future maintenance

Can see what has happened in the past

Allows management to budget and track expenses

Frontend

Login Screen

- Allows users to login without a password

Dashboard

- Shows what tasks need to be done during the day and how much equipment is available. It also shows various budget information
- Features:
  - Shows the time the mowers mowed the greens, tees, approaches, rough, and fairways.
  - Notifies the user for overdue tasks or equipment shortages.
  - Can navigate to other sections

Job Board

- Allows for the management to assign work to each of the crew members
- Features:
  - Has a form to create a task and assign it to a given person
  - Displays all tasks for the day and if they have been completed or not

Special Task Tracker

- Tracks the last time chemicals have been applied, how much clip was obtained the last time the greens were mowed, and where areas of interest (wet areas, animal damage, moss) are located.
- Features:
  - Input fields for the name of the topic and when it was cited
  - Notes for if you want to increase or decrease the clip size and how high the mowers were

Budget Tracker

- Tracks financial data and shows where the expenses are going
- Can create various charts to visualize where the money is going

Equipment Tracker

- Tracks the status of equipment, how much of it is in the repair shop, and if any is being ordered.
- Features:
  - A list to display the quantity and status of the equipment
  - Notes to say if more are being ordered or other information

Backend

Server Implementation

The server will use Node.js and Express.js with these routes:

- POST /api/login: Log in
- POST /api/task: Record a new maintenance task.
- GET /api/tasks: Retrieve a list of past and upcoming tasks.

- POST /api/chemical: Log a chemical application.
- GET /api/reports: Fetch data for financial and maintenance reports.
- POST /api/equipment: Update equipment inventory or maintenance status.

2. Middleware:
    - Validation: Ensure input data is complete.
    - Error Handling: Return error messages for invalid inputs

Database Structure

The database will use MongoDB with Mongoose for object modeling. Key document types include:

1. User Document
    - username: String, unique identifier for the user.
    - tasks: Array of references to Task documents.
2. Task Document
    - area: String, the area being maintained
    - activity: String, the type of activity
    - date: Date, when the activity was completed.
    - notes: String, additional information.
3. Special Task Document
    - name: String, name of the task
    - applicationDate: Date, when it was applied or recorded
    - targetArea: String, area where it was applied or recorded
4. Budget Document
    - category: String, category of the expense
    - amount: Number, the cost of the expense.
    - date: Date, when the expense occurred.
5. Equipment Document
    - name: String, name of the equipment.

    - status: String, current status
    - quantity: Number, total quantity available.

Additional Modules

- mongoose: For database interaction.
- chart.js: For generating reports and graphs.

Timeline

By Wednesday, December 4

- Set up project boilerplate, including Express and MongoDB
- Implement basic routes for login and task creation
- Develop the frontend for login and dashboard
- Complete backend routes for task management and scheduling

By Friday, December 13

- Integrate API calls with frontend
- Add reports visualization
- Conduct testing and fix bugs
- Final testing