Assignment Tracker – Project Plan

1. Project Overview

The goal of this project was to design and develop a web-based Assignment Tracker application that allowed students to organize and manage their school assignments. The system was intended to provide users with the ability to add assignments, view them in a structured list, update details when needed, and delete tasks that were completed or no longer required.

The application was planned to be built using:

- Node.js and ExpressJS for backend logic

- MongoDB and Mongoose for data storage

- EJS templating for the user interface

- Bootstrap for styling

The final objective was to deploy the application online and make it accessible through a live URL.

---

2. Purpose and Problem Being Solved

Students often struggle to keep track of multiple deadlines within many of their courses. This project was planned to solve the problem of:

- Forgetting due dates

- Losing track of priorities

- Lacking a centralized place to manage coursework

The Assignment Tracker was intended to provide a simple, accessible tool for managing academic tasks in one place.

---

3. Target Users

The target users were primarily high school, college and university students who needed a straightforward digital solution to track their assignments and deadlines.

---

4. Planned Features

The following features were planned before development began:

Core Functionality (CRUD)

1. Create a new assignment with fields:

   - Course

   - Title

   - Description

   - Due Date

   - Priority

2. View all assignments in a list format

3. Edit an existing assignment

4. Delete an assignment with a confirmation message

Additional Planned Functionality

- Validation to prevent entering invalid data

- A clean home page acting as a landing page

- A shared header and footer for consistency

- Visual differentiation using Bootstrap styling

---

5. Technology Stack

Component	Planned Technology
Backend	Node.js, ExpressJS
Database	MongoDB, Mongoose
Frontend	EJS Templates, Bootstrap CSS
Deployment	Cloud hosting platform (Render)
Version Control	Git & GitHub

6. Database Plan

A single MongoDB collection named "assignments" was planned, containing the following fields:

Field	Type	Purpose
course	String	Course name
title	String	Assignment title
description	String	Additional details
dueDate	Date	Deadline
priority	String	High/Medium/Low

7. Planned UI Layout

Visual Mockup (Concept)

[Header Navigation]

[Home Page]
Welcome message
Button: View Assignments

[Assignments Page]
Table of assignments:
Course | Title | Due Date | Priority | Actions

[Add/Edit Assignment Page]
Form inputs
Save/Cancel buttons

[Footer]
Application name and year

(You can insert screenshots of your final app here to serve as the "visual" requirement.)

8. Development Timeline (Planned)

Task	Planned Time
Project planning and idea selection	1 day
Setting up Express project	1 day
Database setup and Mongoose model	1 day
CRUD routes and controllers	2 days
UI and Bootstrap styling	2 days
Testing and validation	1 day
Deployment and GitHub setup	1 day

9. Expected Outcome

The planned outcome was a fully functional, deployed web application that allowed users to manage assignments through a clean interface, supported full CRUD operations, and stored data securely in MongoDB.

10. Success Criteria

The project is considered successful if:

- The app performed all CRUD operations

- It had a professional Bootstrap design

- It included a home page, shared header, and footer

- It used .env for security

- It was hosted online and accessible via a public link

- The repository was public with version control history