Project X Plan
By Mila Pulvirenti

TO-DO LIST APPLICATION

Why?

Creating this application will not only fulfill my project requirements but will also serve as a tool in organizing my own tasks and projects. For example, it will help me to break down tasks into stages and set deadlines as well as help me keep track of interactions through creating separate tasks for collecting feedback and implementing changes.

Features:

Basic To-Do List Functionality

- Add, edit and delete tasks.
- Mark tasks as completed.
- Filter tasks based on status (all, active, completed)

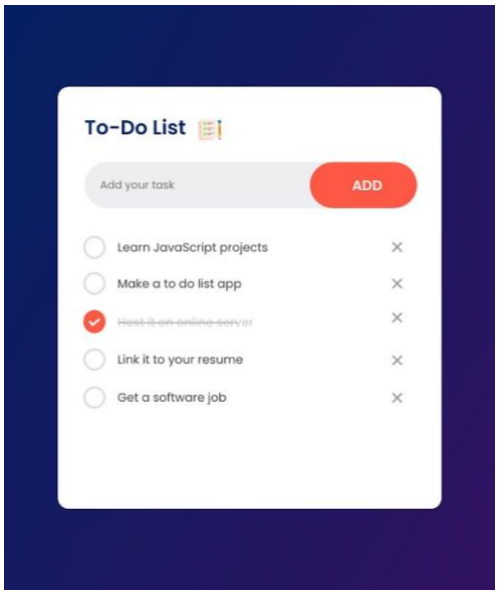
Use Interface Design

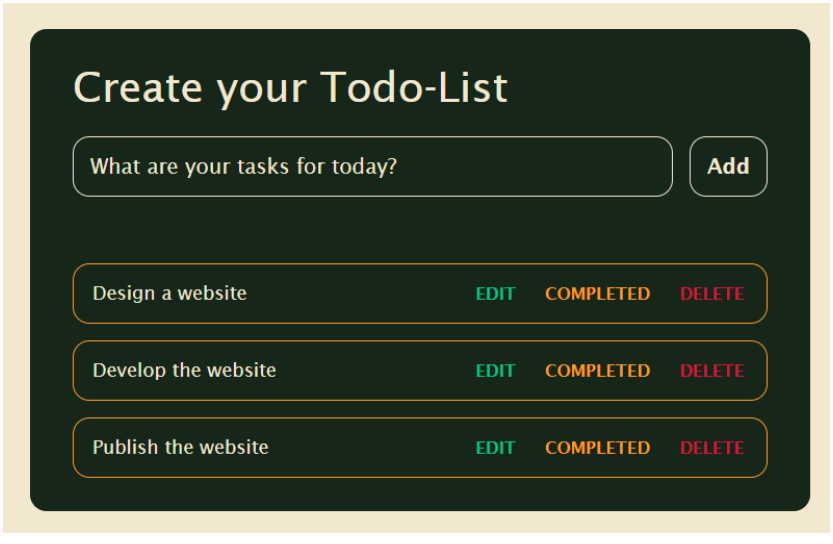
- Use HTML for basic structure.
- Style with CSS to make an appealing design.
- Functionality with JavaScript.

Possible Other Feature Ideas

- Local Storage (Saving tasks in browsers local storage so that they persist across all sections)
- Due Dates (Allow user to add due dates and order tasks by date)
- Categories (Organize tasks by category)
- Responsive Design

Possible Design Inspiration:





CMD METHODS

1. Context Mapping (FIELD)

- Understand the context and needs of potential users by researching existing to-do list applications.
- Conduct interviews or surveys to gather user needs and preferences.

LEARNING OUTCOMES:
- Professional Skills: Understanding the user context and gathering information.

DELIVERABLES:
- User research report

2. Personas (LIBRARY)

- Create personas based on user research to represent different types of users and their goals.

LEARNING OUTCOMES:
- Interactive Media Products: Designing with specific user needs in mind.

DELIVERABLES:
- Persona profiles

3. Wireframing (WORKSHOP)

- Design wireframes to outline the basic structure.

LEARNING OUTCOMES:
- Interactive Media Products: Visualizing the layout.

DELIVERABLES:
- Wireframe sketches and digital wireframes

4. Prototyping (LAB/WORKSHOP)

- Develop a high-fidelity prototype using Figma.

LEARNING OUTCOMES:
- Iterative Design: Improving the prototype based on feedback.

DELIVERABLES:
- High-fidelity interactive prototype

5. User Testing (LAB)

- Conduct usability testing sessions to gather feedback on the prototype.
- Observe and record user interactions to identify pain points and areas for improvement.

LEARNING OUTCOMES:
- Iterative Design: Using feedback to refine and improve the application.

DELIVERABLES:
- Usability test reports

6. Card Sorting

- Use card sorting to organize and categorize tasks within the application (e.g., due dates, categories).

LEARNING OUTCOMES:
- Interactive Prototyping: Ensuring the application is intuitive and user-friendly.

DELIVERABLES:
- Card sorting analysis report.
- Improved task organization scheme.

7. Mood Boarding (WORKSHOP)

- Create a mood board to define the visual style and aesthetic of your application.

Learning Outcomes:
- Interactive Prototyping: Establishing a consistent visual design

DELIVERABLES:
- Mood board

8. Scenario writing (FIELD)

- Write scenarios to describe how different personas will interact with the application.

LEARNING OUTCOMES:
- Interactive Prototyping: Understanding user journeys and interactions.

DELIVERABLES:
- Scenarios and user journey maps.

ACTIVITIES AND LEARNING OUTCOMES

1. Planning and Research

- Conduct context mapping to understand user needs.
- Create personas based on research.

LEARNING OUTCOMES:
- Professional Skills: Understanding user context and needs.

DELIVERABLES:
- User research report.
- Persona profiles.

2. Prototyping

- Create wireframes to outline the application structure.
- Develop a mood board for the design aesthetic.

LEARNING OUTCOMES:
- Interactive Media Products: Designing user interface prototypes and user journeys.
- Iterative Design: Refining designs based on feedback.

DELIVERABLES:
- Wireframes.
- Mood board.
- User journey maps.

3. Development

- Set up the project structure with HTML, CSS, and JavaScript files.
- Implement functionality and features.
- Use card sorting to organize tasks within the application.

LEARNING OUTCOMES:
- Development: Building the application with HTML, CSS, and JavaScript.

DELIVERABLES:
- Functional to-do list application.
- Card sorting analysis report.

4. User Testing and Feedback

- Conduct user testing sessions to gather feedback.
- Analyse feedback and identify areas for improvement.

LEARNING OUTCOMES:
- Iterative Design: Using feedback to improve design and functionality.
- Professional Skills: Conducting and analysing user tests.

DELIVERABLES:
- Usability test reports.

TIMELINE

Week 1: Planning and Prototyping
DAY 1-2: Planning and Research
- Conduct context mapping and create personas.

DAY 3-4: Wireframing and Mood Board
-Design wireframes and develop a mood board.

DAY 5: User Testing with Wireframes
-Conduct user testing sessions with wireframes.
-Collect and analyse feedback.

DAY 6-7: Initial Development
-Make adjustments to wireframes based on feedback.
-Set up project structure and begin implementing core functionality.

Week 2: Development and Iteration
DAY 8-10: Continue Development
-Complete core functionality and implement advanced features.
-Perform card sorting to organize tasks.

DAY 11-12: User Testing with Prototypes
-Conduct user testing sessions with the interactive prototype.
-Collect and analyse feedback.

DAY 13-14: Iteration
-Make improvements based on user feedback.
-Test and refine features.

Week 3: Finalization

DAY 15-16: Optimization and Documentation
-Optimize and refactor code.
-Create user and technical documentation.

DAY 17-21: Final Touches
-Final testing and bug fixing.