**Project Plan: Temperature Converter**

1. **Project overview**

Build a simple website that converts temperatures between Celsius, Fahrenheit, and Kelvin.

1. **Timeline**

Estimated project duration: **2 weeks**

1. **Milestone and Tasks**

•**Task 1.1: Define project scope & requirements**

Time: 0.5 day

Deliverable: Project Requirement Document

• **Task 1.2: Set up project repository and development environment**

Time: 1 day

Deliverable: Initialized Git repository with initial files (README, license, etc.)

• **Task 1.3: Design the basic UI layout and wireframes**

Time: 1 day

Deliverable: UI Wireframes and Design Draft

**Milestone 2: Backend Development**

• **Task 2.1: Set up server with basic routing and endpoints**

• Time: 2 days

• Deliverable: Server running with basic routing for adding and fetching tasks

• **Task 2.2: Implement database for storing tasks**

• Time: 2 days

• Deliverable: Database schema, working database connection

**Milestone 3: Frontend Development**

• **Task 3.1: Build basic user interface (UI) components**

• Time: 2 days

• Deliverable: Static UI components (HTML, CSS)

• **Task 3.2: Connect UI to backend API (fetching data and displaying tasks)**

• Time: 2 days

• Deliverable: Functional connection to API, tasks shown in the UI

• **Task 3.3: Implement convert functionality on the frontend**

• Time: 1.5 days

• **Task 3.4: Implement basic form validation (e.g., empty task name)**

• Time: 0.5 day

• Deliverable: Validation logic for inputs

**Milestone 4: Testing & Bug Fixing**

• **Task 4.1: Write unit tests for backend functionality**

• Time: 1 day

• Deliverable: Backend unit tests with test coverage

• **Task 4.2: Write end-to-end tests for frontend functionality**

• Time: 1 day

• Deliverable: Frontend end-to-end tests

• **Task 4.3: Fix identified bugs and improve error handling**

• Time: 1 day

• Deliverable: Bug-free system, improved error handling messages

**Milestone 5: Deployment & Final Review**

• **Task 5.1: Deploy the application to a cloud service (e.g., Heroku, Netlify)**

• Time: 1 day

• Deliverable: Deployed application URL

• **Task 5.2: Final review and project documentation**

• Time: 1 day

• Deliverable: Complete project documentation (setup guide, API documentation, etc.)

• **Task 5.3: Collect feedback from stakeholders and make final adjustments**

• Time: 0.5 day

• Deliverable: Final version of the application with feedback incorporated

**4. Deliverables**

1. **Project Requirements Document**

• Details of the app's functionality and design.

2. **Source Code**

• Full project source code (backend, frontend, and any configurations).

3. **UI Wireframes/Design**

• Visual representation of the app's user interface.

4. **Working Web Application**

• Deployed, live version of the To-Do List web application.

5. **API Documentation**

• Documentation for the API endpoints used in the project.

6. **Test Suite**

• Unit tests for backend and end-to-end tests for the frontend.

7. **Final Project Report**

• A summary of the project, any challenges faced, and how they were overcome.

**5. Risks and Mitigation Plans**

• **Risk:** Lack of time for frontend refinement.

**Mitigation:** Focus on core functionality first, and refine UI after the features are complete.

• **Risk:** Potential issues with database integration or API bugs.

**Mitigation:** Regular code reviews, testing, and debugging to identify and resolve issues early.

**6. Project Schedule Overview**

**Week Task Duration**

1 Project setup & backend work 4 days

1 Frontend development 4 days

2 Testing & bug fixing 3 days

2 Deployment & documentation 3 days

This plan can be adapted based on the complexity of the project or if you're working with a team.