Week 1 Assignment Rubric:

Product Requirements Document & Initial Prototype

**Total Points: 100**

# Overview

Define your product vision and create a simple but functional prototype. You'll document your idea in a Product Requirements Document (PRD) and build a basic website that demonstrates the core concept.

## What is a PRD?

A **Product Requirements Document** is a business document that explains *what* you're building and *why*, without diving into technical implementation details. A classic PRD structure includes:

* **Problem Statement:** What problem are you solving?
* **Target Users:** Who will use this product?
* **Goals & Success Metrics:** What does success look like?
* **User Stories:** What will users be able to do?
* **MVP Features:** What's included in the first version?
* **Future Roadmap:** What comes next?
* **Open Questions:** What needs to be decided?

# Part 1: Product Requirements Document (55 points)

## 1. Executive Summary (8 points)

**Requirements:**

* Clear, compelling product name
* 2-3 sentence elevator pitch explaining what the product does
* Brief statement of the core problem being solved

**Evaluation:**

* **8 points:** Concise, compelling, immediately understandable
* **6 points:** Clear but could be more engaging or focused
* **4 points:** Present but unclear or too verbose
* **0 points:** Missing or incomprehensible

## 2. Problem Statement & Opportunity (10 points)

**Requirements:**

* Clearly articulate the problem your product solves
* Explain why this problem matters (who's affected, frequency, impact)
* Describe the current alternatives and their shortcomings

**Evaluation:**

* **10 points:** Compelling problem with clear pain points and well-researched context
* **7 points:** Problem is clear but lacks depth or context
* **4 points:** Problem is vague or unconvincing
* **0 points:** Missing or doesn't describe an actual problem

## 3. Target Users & User Personas (8 points)

**Requirements:**

* Define 1-2 specific user personas (who they are, their needs, their context)
* Explain why these users need this product

**Evaluation:**

* **8 points:** Specific, well-defined personas with clear motivations
* **6 points:** Personas are present but somewhat generic
* **3 points:** Vague or unrealistic user descriptions
* **0 points:** Missing or unclear who the users are

## 4. MVP Feature Specifications (12 points)

**Requirements:**

* List 3-5 core features for the Minimum Viable Product
* For each feature, include:

• Feature name

• User story format: "As a [user], I want to [action] so that [benefit]"

• Acceptance criteria (what defines "done")

**Evaluation:**

* **12 points:** Features are well-defined, appropriately scoped for MVP, with clear user stories and criteria
* **9 points:** Features are clear but missing some user stories or acceptance criteria
* **6 points:** Features listed but lack detail or are poorly scoped (too ambitious or too trivial)
* **3 points:** Features are vague or missing key components
* **0 points:** Missing or inappropriate feature set

## 5. Future Roadmap (7 points)

**Requirements:**

* Outline 4-6 features planned for future iterations (Weeks 2-6)
* Group them logically (e.g., "Week 2-3 Features", "Week 4-5 Features")
* Brief description of each future feature

**Evaluation:**

* **7 points:** Logical progression, features build on each other, realistic scope
* **5 points:** Future features present but organization or scope could be improved
* **2 points:** Features are too vague or don't build coherently
* **0 points:** Missing or completely unrealistic

## 6. Success Metrics (5 points)

**Requirements:**

* Define 2-4 measurable ways to determine if the product is succeeding
* Should be specific (not just "users love it")

**Evaluation:**

* **5 points:** Specific, measurable, relevant metrics
* **3 points:** Metrics present but somewhat vague
* **1 point:** Metrics are too generic or unmeasurable
* **0 points:** Missing

## 7. Open Questions (5 points)

**Requirements:**

* List 3-5 unresolved business/product decisions that need answers
* Examples: "Should we charge for premium features?", "Which user segment should we focus on first?", "What's the ideal onboarding flow?"
* Should be thoughtful questions that impact product direction (**NOT** technical questions like "Which database should we use?")

**Evaluation:**

* **5 points:** Thoughtful, impactful questions that show strategic thinking
* **3 points:** Questions are relevant but somewhat obvious or low-impact
* **1 point:** Questions are too technical or not substantive
* **0 points:** Missing or inappropriate questions

# Part 2: Initial Prototype (30 points)

## 8. Functional Website (15 points)

**Requirements:**

* Built with vanilla HTML, CSS, and JavaScript (no frameworks)
* Website must be functional—buttons should work, forms should respond, interactions should do something meaningful
* Should demonstrate at least ONE core concept from your MVP
* Must be deployed and accessible via Vercel

**Evaluation:**

* **15 points:** Fully functional, demonstrates core concept clearly, clean execution
* **11 points:** Mostly functional with minor issues, concept is evident
* **7 points:** Partially functional or concept unclear
* **3 points:** Minimal functionality or mostly non-functional
* **0 points:** Missing, not deployed, or completely non-functional (no-op code)

## 9. Code Quality & Organization (8 points)

**Requirements:**

* Code is organized and readable
* HTML is semantic and properly structured
* CSS is organized (not all inline)
* JavaScript is in separate file(s) or properly organized in <script> tags
* No console errors in browser

**Evaluation:**

* **8 points:** Well-organized, clean code structure, no errors
* **6 points:** Organized but could be cleaner or has minor errors
* **3 points:** Disorganized or significant errors
* **0 points:** Chaotic code or completely broken

## 10. Visual Design & Polish (7 points)

**Requirements:**

* Website has intentional styling (not default browser styles)
* Layout is clean and usable
* Basic responsive considerations (looks okay on different screen sizes)
* Represents your product concept visually

**Evaluation:**

* **7 points:** Polished, intentional design that reflects the product
* **5 points:** Styled but basic or inconsistent
* **3 points:** Minimal styling effort
* **0 points:** No styling or completely unusable

# Part 3: Repository & Documentation (15 points)

## 11. Git Repository Setup (5 points)

**Requirements:**

* Public repository created
* Clear README.md at root with:

• Project name

• Brief description

• Link to deployed Vercel site

• Link to PRD

* PRD in /projectDocs/ folder
* Clean repository structure

**Evaluation:**

* **5 points:** Perfect setup with clear README
* **3 points:** Setup correct but README could be better
* **1 point:** Repository exists but poorly organized
* **0 points:** Missing or not public

## 12. Git Commit History (5 points)

**Requirements:**

* Repository shows regular commits throughout the development process
* Each logical change to the app should be committed in separate commits
* Commit messages should be descriptive (not just "update" or "fix")
* **NOT acceptable:** A single commit at the end with all files

**Evaluation:**

* **5 points:** Multiple logical commits showing clear development progression with descriptive messages
* **3 points:** Some commits but could be more granular or messages could be clearer
* **1 point:** Very few commits or poor commit messages
* **0 points:** Single commit with everything, or no meaningful commit history

## 13. PRD Formatting & Professionalism (5 points)

**Requirements:**

* PRD is in markdown format
* Proper markdown formatting (headers, lists, emphasis)
* Professional tone and no major typos
* Easy to read and navigate

**Evaluation:**

* **5 points:** Professional, well-formatted, easy to read
* **3 points:** Formatted but could be cleaner or has typos
* **1 point:** Minimal formatting or hard to read
* **0 points:** Not in markdown or completely unprofessional

# Submission Requirements

Submit to the course portal:

1. Link to your public GitHub repository

2. Link to your deployed Vercel site

3. Both must be accessible at submission time

**Late Penalty:** -10 points per day late

# Grading Summary

| **Component** | **Points** |
| --- | --- |
| Executive Summary | 8 |
| Problem Statement & Opportunity | 10 |
| Target Users & Personas | 8 |
| MVP Feature Specifications | 12 |
| Future Roadmap | 7 |
| Success Metrics | 5 |
| Open Questions | 5 |
| Functional Website | 15 |
| Code Quality & Organization | 8 |
| Visual Design & Polish | 7 |
| Git Repository Setup | 5 |
| Git Commit History | 5 |
| PRD Formatting & Professionalism | 5 |
| **TOTAL** | **100** |

# Tips for Success

* **Use AI effectively:** AI can help brainstorm ideas, structure your PRD, and debug code. But make sure you understand and can explain everything you submit.
* **Start with the problem:** The best products solve real problems. Don't start with a cool technology—start with a pain point.
* **Keep MVP simple:** Your first version should be the simplest possible thing that demonstrates value.
* **Make it work:** A simple, working prototype is better than an ambitious, broken one.
* **Commit early and often:** Make commits as you complete each logical piece of work. Good commit history shows your development process and makes it easier to track down issues.
* **Iterate on your PRD:** First drafts are rarely perfect. Review and refine.

Good luck! 🚀