### **1. Product Name:**

* Aptolize

**2. Team members:**

* **Lokeshwaran B:** Project Lead, [lokeshwaran100@gmail.com]
* **Dharshan S:** Full Stack Developer, [dharshan2457@gmail.com]
* **Adithya NG:** Frontend Developer
* **Akhil:** Smart Contract Developer

### **3. Problem Statement:**

* Many users who hold Aptos are unable to fully utilize their assets due to the lengthy lock-up periods required for staking. This creates a barrier for those with smaller amounts of Aptos who want to participate in staking but don’t want their funds locked for extended periods.
* Additionally, the lack of active staking can lead to increased market volatility, as unstaked assets are more likely to be traded or sold quickly.

### **4. Target Market:**

* Aptolize is aimed at Aptos holders with smaller amounts of the currency, who want to maximize their potential rewards without locking up their funds for 30 days. These users are looking for a way to participate in staking while still maintaining the flexibility to access their funds when needed.

### **5. Core Features:**

* **Flexible Staking:** Users can stake their Aptos without the need for a 30-day lock-up, providing them with liquidity and flexibility.
* **Lottery-Based Rewards:** A lottery system that allows users to win rewards based on luck, giving even small stakers the chance to gain significant rewards.
* **Stake, Unstake, and Claim Rewards:** Simple interfaces for staking, unstaking, and claiming lottery rewards, integrated directly into the platform.
* **Collaborative Quests:** Users can complete tasks to increase their chances of winning, fostering community engagement with partner projects.

### **6. Value Proposition:**

* Aptolize allows users to stake their Aptos without the need for long lock-up periods, ensuring they can access their funds when needed. At the same time, by staking through Aptolize, users contribute to reducing market volatility by keeping a greater portion of the circulating supply staked on the platform.
* With the added benefit of a lottery system, even users with smaller amounts of Aptos have the opportunity to win larger rewards based on luck, making staking more accessible and rewarding for everyone.

### **7. Development Timeline:**

### **Day 1: Design & Initial Setup**

* **Design and Implement Logo for Aptolize**
  + **Resources:** Graphic Designer, Design Tools (Canva)
  + **Milestone:** Completed logo design aligned with Aptolize's brand identity.
* **Create Next.js Landing Page with Tailwind CSS**
  + **Resources:** Frontend Developer, Next.js, Tailwind CSS, Design Specifications
  + **Milestone:** Basic landing page with a responsive design.
* **Add Navigation Bar to Next.js App**
  + **Resources:** Frontend Developer, Navigation Menu Design
  + **Milestone:** Functional navigation bar integrated into the landing page.

### **Day 2: Frontend Development**

* **Create Dashboard Page with Stake, Unstake, and Claim Buttons**
  + **Resources:** Frontend Developer, Tailwind CSS, Next.js
  + **Milestone:** Interactive dashboard page with staking options.
* **Implement Stake Dialog Button Component**
  + **Resources:** Frontend Developer, Component Design
  + **Milestone:** Functional stake dialog integrated with the dashboard.
* **Implement Unstake Dialog Button Component**
  + **Resources:** Frontend Developer, Component Design
  + **Milestone:** Functional unstake dialog integrated with the dashboard.
* **Implement Claim Dialog Button Component**
  + **Resources:** Frontend Developer, Component Design
  + **Milestone:** Functional claim dialog integrated with the dashboard.

### **Day 3: Smart Contract Development & Integration**

* **Create Aptos Smart Contract with Stake, Unstake, and Lottery Functions**
  + **Resources:** Smart Contract Developer, Aptos SDK, Testnet
  + **Milestone:** Deployable smart contract with staking, unstaking, and lottery logic.
* **Integrate Smart Contract Stake Function with Stake Dialog**
  + **Resources:** Frontend Developer, Smart Contract Developer
  + **Milestone:** Fully functional staking mechanism.
* **Integrate Smart Contract Unstake Function with Unstake Dialog**
  + **Resources:** Frontend Developer, Smart Contract Developer
  + **Milestone:** Fully functional unstaking mechanism.
* **Integrate Smart Contract Claim Function with Claim Dialog**
  + **Resources:** Frontend Developer, Smart Contract Developer
  + **Milestone:** Fully functional claim mechanism.

### **Day 4: Final Touches & Testing**

* **Develop Toast Component for Notifications**
  + **Resources:** Frontend Developer, UX Designer
  + **Milestone:** Notification system that informs users of successful or failed transactions.
* **Research Admin Access for Choose Winner Function in Aptos Smart Contract**
  + **Resources:** Smart Contract Developer, Aptos SDK Documentation
  + **Milestone:** Implemented admin function for selecting a daily lottery winner.
* **Final Testing and Review**
  + **Resources:** Entire Team
  + **Milestone:** Ensure all components and smart contract integrations work seamlessly.

### **8. User Interface (UI) Design:**

* [Provide basic wireframes or mockups to visualize the UI.]

### **9. User Experience (UX) Flow:**

* [Describe the user journey and interactions with the MVP.]

### **10. Technical Requirements:**

* [Outline the necessary software, and programming languages.]

### **11. Testing and Validation:**

* [Plan how you will test the MVP and measure its success.]

### **12. Launch Strategy:**

* [Describe how you will introduce the MVP to the market.]

### **13. Feedback and Iteration:**

* [Explain how you will gather user feedback and improve the MVP - this is after the hackathon]

### **14. Risk Assessment:**

* [Identify potential risks and develop mitigation strategies.]

15. **Roadmap**

* [3 months realistic roadmap of the product]