# Project Planning – HealthAI

Date: 25 JUNE 2025

Team ID: LTVIP2025TMID31761

Project Name: HealthAI

Maximum Marks: 5

## Product Backlog, Sprint Schedule, and Estimations (4 Marks)

This section outlines the prioritized product backlog, distributed across planned sprints, with estimated story points and assigned team members.

### Sprint-wise User Stories

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Epic | User Story No. | User Story | Story Points | Priority | Team Member |
| Sprint-1 | UI & Core Chat | USN-1 | As a user, I can ask health questions & get AI replies (using Gemini API) | 3 | High | Member A |
| Sprint-1 | Patient Profile | USN-2 | As a user, I can input and update my patient profile details. | 2 | High | Member B |
| Sprint-1 | Core Setup | USN-3 | As a developer, I can initialize the AI model connection (Gemini API) & basic Streamlit layout. | 3 | High | Member A |
| Sprint-2 | Prediction Module | USN-4 | As a user, I can enter symptoms for disease prediction (using Gemini API). | 3 | High | Member C |
| Sprint-2 | Treatment Module | USN-5 | As a user, I can generate personalized treatment plans (using Gemini API). | 3 | High | Member C |
| Sprint-3 | Health Analytics UI | USN-6 | As a user, I can view interactive health trend charts (Heart Rate, BP, Glucose). | 3 | Medium | Member D |
| Sprint-3 | Analytics Insights | USN-7 | As a user, I can get AI-generated health insights based on my data. | 2 | Medium | Member A |
| Sprint-3 | Session Management | USN-8 | As a user, my chat and profile history persist within a session. | 2 | Low | Member B |
| Sprint-4 | Deployment Prep | USN-9 | As a developer, I can prepare the application for cloud deployment. | 3 | High | Member D |
| Sprint-4 | Documentation | USN-10 | As a team, we can finalize all project documentation (UAT, Tech Stack, etc.). | 2 | Medium | All Members |
| Sprint-4 | IBM Integration (PoC) | USN-11 | As a developer, I can explore initial integration with IBM Granite. | 3 | High | Member C |

### Sprint Schedule

This schedule outlines the planned duration and key milestones for each sprint.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Story Points | Duration | Start Date | End Date | Completed Points | Release Focus |
| Sprint-1 | 8 | 6 Days | 10 Feb | 15 Feb | 8 | Basic UI & Chatbot (Gemini) |
| Sprint-2 | 6 | 6 Days | 16 Feb | 21 Feb | 6 | Prediction & Treatment (Gemini) |
| Sprint-3 | 7 | 6 Days | 22 Feb | 27 Feb | 7 | Health Analytics & Session Mgmt |
| Sprint-4 | 8 | 6 Days | 28 Feb | 04 Mar | TBD | Deployment & IBM Integration PoC |

## Project Tracker, Velocity & Burndown Chart (1 Mark)

### Velocity Calculation:

Average Velocity: Based on completed sprints, Average Velocity = (8 + 6 + 7) / 3 = 7 Story Points per sprint.

This velocity indicates the team's capacity for upcoming sprints.

### Burndown Chart:

A burndown chart will be generated from the project management tool (e.g., Jira, Trello) showing the progress of completed story points against the total work for each sprint. This visual will demonstrate consistent progress and sprint-wise point closure.

Data so far (Example):

• Sprint 1: All 8 points completed as planned.

• Sprint 2: All 6 points completed as planned.

• Sprint 3: All 7 points completed as planned.

• Sprint 4: Progress to be tracked.

## Tools Used:

* Jira/Trello/Asana: For backlog tracking, assigning stories, sprint reviews, and overall project management.
* VS Code: Primary Integrated Development Environment (IDE) for coding.
* Streamlit: For rapid interface-level sprint delivery validation and core UI development.
* Google Gemini API: For simulating AI functionalities during development.
* Plotly: For interactive data visualization in the Health Analytics dashboard.
* Git/GitHub: For version control, collaborative development, and code management.
* Manual Testing: For comprehensive sprint validation and tracking feature completion.

## Conclusion

The HealthAI project follows Agile principles, emphasizing iterative delivery and continuous feedback. Sprint planning aligns with feature-based prioritization, ensuring that core functionalities are delivered progressively. The use of robust tools and a structured approach facilitates efficient development and prepares the application for full-scale deployment and future enhancements.