# Agile Artifacts for the Community Health Portal

## 1. Product Backlog with Prioritized User Stories

The product backlog aligns with the user personas and challenges described in the document. The user stories are crafted to address specific needs and requirements.

|  |  |  |
| --- | --- | --- |
| Priority | User Story | Acceptance Criteria |
| High | As a community health worker, I want access to multilingual educational resources, so I can provide preventive care to underserved communities. | - The system supports at least three languages. - Easy access to educational materials via a user-friendly interface. |
| High | As a general healthcare provider, I want an integrated patient management system, so I can streamline teleconsultations and follow-ups. | - Providers can view patient history in a unified dashboard. - Teleconsultation logs are automatically stored in patient records. |
| High | As a young adult, I want quick appointment booking and notifications, so I can fit healthcare into my busy schedule. | - Users can book, reschedule, or cancel within 5 clicks. - Notifications for upcoming appointments and changes are delivered promptly. |
| Medium | As an elderly user, I want an easy-to-navigate interface, so I can manage my appointments and access telehealth services independently. | - Simplified navigation with larger icons and text. - Accessible design with screen reader compatibility. |
| Medium | As a caregiver, I want to manage appointments and health records for my dependent, so I can coordinate their care effectively. | - Caregivers can switch between their account and dependent's account. - Appointments and records are displayed in a unified timeline. |
| Medium | As a mental health support seeker, I want secure and private teleconsultations, so I can discuss my concerns without fear of judgment. | - Secure end-to-end encrypted consultations. - Anonymous login options for sensitive mental health queries. |

## 2. Sprint Plan and Detailed Task Breakdown for the First 2-3 Sprints

### Sprint 1: Foundation Setup

Goal: Establish the core platform infrastructure and basic functionality for user authentication and dashboards.

Tasks:

1. 1. Set up hosting and database infrastructure for user and appointment management (2 days).  
   2. Implement user authentication (registration, login, and role-based access) (4 days).  
   3. Develop the initial version of the patient dashboard, with placeholders for health records and appointments (3 days).  
   4. Conduct unit testing and integration testing for authentication (2 days).

Deliverables:

* - Functional user authentication system.  
  - Basic dashboard interface with placeholders.

Dependencies:

* - Secure hosting environment, UI design assets.

### Sprint 2: Appointment Management

Goal: Enable appointment booking, management, and notifications.

Tasks:

1. 1. Design and develop appointment booking features, including availability checking and time slot selection (4 days).  
   2. Implement notification and reminder systems (email/SMS) (3 days).  
   3. Integrate appointment rescheduling and cancellation functionality (3 days).  
   4. QA testing and user feedback implementation for the appointment module (2 days).

Deliverables:

* - Real-time appointment booking and management system.  
  - Notification and reminder integration.

Dependencies:

* - Notification API, design specifications for the appointment UI.

### Sprint 3: Telehealth and Accessibility Features

Goal: Launch teleconsultation capabilities and ensure accessibility compliance.

Tasks:

1. 1. Develop teleconsultation modules with text-based chat and video call integration (5 days).  
   2. Conduct accessibility testing and implement changes to meet compliance standards (3 days).  
   3. Build and integrate a caregiver mode for managing dependent accounts (3 days).  
   4. Perform end-to-end testing of teleconsultation and caregiver features (2 days).

Deliverables:

* - Functional teleconsultation module.  
  - Accessibility-compliant user interfaces.  
  - Caregiver mode integrated with the platform.

Dependencies:

* - Video conferencing API, accessibility guidelines.

## 3. Sprint Retrospective and Burndown Charts

### Sprint Retrospective

At the conclusion of each sprint, a retrospective meeting will be conducted to review:

* - What went well: Efficient completion of tasks, effective team communication, and early identification of blockers.  
  - What could be improved: Addressing bottlenecks in dependency management, enhancing collaboration for cross-functional tasks.  
  - Actionable items for improvement: Allocate more resources for testing and documentation, establish clearer timelines for external dependencies.

### Burndown Charts

Burndown charts will be maintained for each sprint to visualize progress and ensure timely completion. The chart will track:

* - X-Axis: Days in the sprint.  
  - Y-Axis: Remaining effort (in story points or hours).

1. The burndown chart will be updated daily with the following:  
   1. Remaining Effort: The number of pending tasks or story points.  
   2. Trend Line: Represents ideal progress if all tasks are completed evenly over the sprint.