Sprint 1 Planning Document
Project: Community Service App for the Government of Ontario
Team: PRJ666 – Team 3
Sprint Duration: Weeks 1–3 (3 Weeks)

Sprint Goal
The primary objective of Sprint 1 is to lay the technical groundwork for the Community Service App by setting up the development infrastructure and implementing the complete resident authentication module. This includes establishing a robust, secure authentication mechanism for residents, session token handling, password recovery via email, and enforcing role-based access control for clerks and administrators. Additionally, this sprint ensures all foundational systems such as GitHub for version control, Vercel for hosting, and MongoDB Atlas for data storage are configured, connected, and operational. These components will serve as the baseline for subsequent feature development.

Major Sprint Adjustments:

- **Reprioritization and reassignment of issue ownership** based on team availability and domain knowledge.
- **Addition of a Main Landing Page UI**, which was not originally planned in the SRS but was added during the sprint for improved app structure and user experience.
- **UI revamp for login, registration, Forgot Password and reset pages** to align with a unified and responsive design system.

Sprint Team Members & Assigned Tasks

Name	Assigned Tasks
Vrundaben Vijaykumar Patel	CS-004: Forgot Password Flow, CS-005: Role-Based Access
Sanskar Parakhlal Pardesi	CS-011: Project Infrastructure & Setup, CS-006: Clerk/Admin Login
Nadi Aung Lin	CS-002: User Login, CS-003: Session Management
Abhi Mansukhbhai Chakrani	CS-001: User Registration

Sprint Scope
This sprint aims to deliver the backend and frontend foundations for the app, focusing on the following:

- Implementing resident registration and login with email and password
- Ensuring session management using secure token mechanisms such as JWT

- Allowing users to securely reset forgotten passwords via email tokens
- Applying Role-Based Access Control (RBAC) to differentiate resident, clerk, and admin access levels
- Enabling clerk and admin login with proper role verification and redirection
- Setting up a functional development environment: GitHub repository, project board, MongoDB Atlas, Vercel hosting, and code formatting standards using ESLint and Prettier

Sprint Backlog Items

ID	Title	Description	Assignee	Effort (SP)
CS-001	User Registration	Implement resident sign-up with email/password and role logic	Abhi M. Chakrani	5
CS-002	User Login	Issue/store token, manage session expiration and access	Sanskar P. Pardesi	3
CS-003	Session Management	Middleware to ensure protected route access	Vrundaben V. Patel	3
CS-004	Forgot Password Flow	Secure reset via email token and update flow	Vrundaben V. Patel	8
CS-005	Role-Based Access Control	Restrict feature access based on user roles	Vrundaben V. Patel	5
CS-006	Clerk and Admin Login	Login for clerk/admin users with role redirection	Sanskar P. Pardesi	3
CS-011	Project Infrastructure Setup	GitHub, MongoDB, Vercel setup with CI and lint config	Sanskar P. Pardesi	4

Sprint Task Breakdown
CS-001: User Registration

- Create and validate registration form with required fields (email, password, optional phone)
- Hash passwords using a secure hashing algorithm before storing
- Save the new user record with default role: resident
- Redirect user to the resident dashboard upon successful registration

CS-002 & CS-003: Login and Session Management

- Verify user credentials and handle login requests
- Generate JWT token on successful login
- Store JWT in localStorage or cookies
- Use middleware to verify token presence and expiration before route access
- Redirect user to login if session is missing or expired

CS-004: Forgot Password Flow

- Add "Forgot Password?" link on login form
- Generate reset token and email it to the registered user
- Validate token on password reset page
- Hash and update the new password in MongoDB

CS-005: Role-Based Access Control (RBAC)

- Check user role after login and dynamically render appropriate UI components
- Prevent users from accessing unauthorized routes using frontend middleware
- Configure backend endpoints to verify roles before processing sensitive actions

CS-006: Clerk/Admin Login

- Extend login feature to support clerk and admin credentials
- Redirect clerks to the clerk dashboard and admins to the admin panel
- Prevent clerk/admin users from accessing resident-only features

CS-011: Project Infrastructure Setup

- Initialize GitHub repo with appropriate folders and README
- Create Vercel project and deploy the base Next.js app
- Connect and configure MongoDB Atlas for data operations
- Set up .env file and ensure environment security
- Integrate ESLint and Prettier with project to enforce consistent formatting
- Optionally configure GitHub Actions for automatic deployments

---

Sprint Acceptance Criteria

- Resident registration and login functionality is working with form validations

- JWT session tokens are securely issued and validated across all protected routes
- Users can securely reset forgotten passwords via token-based email links
- Clerk and Admin login is operational with accurate role-based redirection
- RBAC middleware protects route access and hides unauthorized UI elements
- GitHub repository is initialized with branch structure and commit history
- MongoDB Atlas is live and connected to deployed app on Vercel
- Codebase follows Prettier formatting and ESLint rules

Sprint Deliverables

1. Community Service App GitHub repository with clean initial commits and collaboration setup
2. Live deployment on Vercel with MongoDB Atlas as backend
3. Fully implemented resident registration, login, and logout functionality
4. Middleware-based session management and route protection
5. Secure password recovery system using tokenized email links
6. Clerk and Admin login capabilities with dashboard redirection
7. Role-aware dynamic interface (UI adapts based on user role)
8. Pages for registration, login, and password reset (responsive and functional)

Definition of Done (DoD)

- All assigned tasks are implemented, committed, and merged to the development branch
- All new code is tested manually or via automated tests
- Sensitive information such as passwords and tokens are encrypted
- Session and access control logic passes test scenarios
- Code is linted and auto-formatted before push
- Project is accessible via deployed URL and connected to live database

---

Sprint Risks & Mitigation

Risk	Mitigation Strategy
Email sending may fail during password reset	Use Mailtrap or mock SMTP service for testing in early development
Frontend and backend integration delays	Allocate time for interface contracts and pair programming sessions
Vercel deployment or environment config issues	Schedule buffer during Week 1 to resolve infrastructure-related blockers

Sprint Timeline

Week	Key Activities
1	Project setup: GitHub, Vercel, MongoDB; start CS-001 and CS-011 tasks
2	Complete CS-001; begin CS-002, CS-003, CS-005
3	Finalize CS-004 and CS-006; perform sprint-wide testing and UI refinements

Sprint Review & Demo Goals

- Live demo of user registration and login functionality with protected dashboard routing
- Demonstration of forgot password flow with email token, reset, and login confirmation
- Display of UI rendering based on user roles (resident, clerk, admin)
- Code walkthrough in GitHub showing commits, branches, and formatting compliance
- Vercel deployed app with working MongoDB backend integration