**Product Requirements Document: CivicVigilance**

**1. Introduction & Vision**

* **Product Name:** **CivicVigilance**
* **Mission:** To empower citizens to become active participants in improving their communities. We aim to bridge the communication gap between the public and civic authorities by providing a simple, transparent, and effective platform for reporting and tracking local issues.
* **Elevator Pitch:** **CivicVigilance** is a mobile app that lets you report civic problems like potholes, garbage, or broken streetlights by simply snapping a photo. The app automatically geo-tags the issue and helps you notify the correct authorities on social media, turning every citizen into a community watchdog.

**2. Target Audience**

* **Primary:** Daily commuters, resident welfare association (RWA) members, and socially active citizens who are keen to see improvements in their locality.
* **Secondary:** Civic authorities, NGOs, and local journalists who can use the platform's data to identify and address problem areas.

**3. User Stories**

* **As a driver,** I want to quickly report a dangerous pothole on my route so that the municipal corporation can fix it and prevent accidents.
* **As a concerned resident,** I want to highlight a garbage pile-up in my neighborhood to get it cleaned and maintain local hygiene.
* **As a user,** I want to see what issues others in my area are reporting so I can upvote them for greater visibility and feel a sense of community.

**Screen-by-Screen Scope of Work**

**Screen 1: Onboarding & Authentication**

This covers the user's first interaction with the app, focusing on secure and easy access.

**Screens:**

1. **Login Screen:** The entry point for existing users.
2. **Create Account Screen:** For new user registration.
3. **Forgot Password Screen:** To help users reset their password.

**Functionality:**

* **Email/Password Login:** Standard login form with validation for email format and password fields.
* **Social Logins (OAuth 2.0):** One-tap sign-in/sign-up using **Google, Apple, Facebook, and X (Twitter)**. This simplifies registration and reduces login friction.
* **User Registration:** The "Create Account" form will collect Name, Email, DOB (for demographic analysis, optional), Location (optional), and a securely hashed Password.
* **Password Recovery:** A standard "Forgot Password" flow that sends a secure reset link to the user's registered email.

**Screen 2: Home Feed**

This is the main dashboard where users discover, engage with, and validate reported issues.

**Functionality:**

* **Dynamic Feed:** A vertically scrolling list of issue cards. The feed should be sortable by "Trending" (most upvotes), "Newest," and "Nearby."
* **Issue Card:** Each card will display:
  + A concise title/description (e.g., "Pothole near Maple St.").
  + An image of the issue (if provided).
  + Key metrics: **Upvote/Downvote counts, Comment count, and a Share button**.
* **User Interaction:**
  + **Voting:** Users can upvote or downvote an issue to influence its priority and visibility. A user can only cast one vote (or change their existing vote).
  + **Commenting:** Tapping the comment icon takes the user to the Post Detail screen to view and add comments.
  + **Sharing:** Allows users to share the link to the **CivicVigilance** post on other platforms.
* **Report New Issue:** A prominent floating action button (+) navigates the user to the "Report Issue" screen.

**Screen 3: Report Issue Flow**

This is the core feature of the app, designed to be quick and intuitive.

**Functionality:**

* **Step 1: Capture Evidence:**
  + The user can take a photo using the in-app camera or upload one from their device's gallery.
  + The app will request **Camera and Location permissions** upon first use.
* **Step 2: Add Details:**
  + **Issue Type:** Users must select a category (e.g., Pothole, Garbage, Streetlight, Water Leak, Other) from a predefined list of tags.
  + **Geolocation:** The app automatically captures the photo's **GPS coordinates (Latitude/Longitude)** and uses a reverse geocoding service to fetch the human-readable address. The user can adjust the pin on a map for accuracy.
  + **Description:** A simple text field for a short description.
  + **Tag Authorities:** The app will suggest relevant authority Twitter handles (e.g., @municipal\_corp, @trafficpolice) based on the issue's location and type. This requires a backend database that maps geographical areas to the correct authorities.
* **Step 3: Preview & Share:**
  + **Post Preview:** The app generates a formatted text post, combining the description, address, location link, and tagged authorities.
  + **Sharing Options:**
    - **Post to X (with Image):** The primary call-to-action. This opens the X (Twitter) app with the image and the pre-filled text, ready to be posted.
    - **Share Post (Image + Text):** Opens the native OS share sheet for sharing on other platforms like WhatsApp, Instagram, etc.
    - **Copy Text:** Copies the generated text to the clipboard.

**Screen 4: Profile & Settings**

This screen allows users to manage their activity and app preferences.

**Functionality:**

* **User Information:** Displays the user's name and other public profile details.
* **My Reports:** A tab or section that shows a list of all the issues the user has personally reported. This allows them to track the status or engagement on their posts.
* **Settings:** Access to app settings, which could include:
  + Notification preferences.
  + Managing linked social accounts.
  + Privacy Policy and Terms of Service.
  + A **Logout** button.