Deliverable 1 (D1)

Campus(NC/OC)	OC
Project ID	
Project Title	StreetLight - “No civic hazard should remain in dark”
Team Lead Roll Num	BITF22M015
Supervisor	Sir Adeel Nisar

Capstone Project Deliverable-1

StreetLight

“No civic hazard should remain in dark”

Department - Campus	Roll Number	Name	Signatures
IT-OC	BITF22M006	Mariam Qadeem	
IT-OC	BITF22M011	Areeba Tahir	
IT-OC	BITF22M015	Kinz-ul-Eman	
IT-OC	BITF22M044	Shezonia	

Supervisor Name	Dept	Email	Primary/ Secondary	Signatures
Adeel Nisar	IT	adeel.nisar@pucit.edu.pk	Primary	

Department of Information Technology

Faculty of Computing & Information Technology
University of the Punjab

Table of Contents

1. Problem Statement

Problem Definition

In Pakistan, citizens face daily civic issues like garbage accumulation, potholes, broken roads, and faulty streetlights that affect their safety and quality of life, yet there is no effective platform to report these problems. Current systems rely on phone calls, scattered emails, or outdated methods that are slow and unaccountable, leaving citizens unaware of which department handles their issue or if action is being taken. Meanwhile, municipal departments lack proper data to identify recurring problems or predict risks, forcing them into reactive crisis management rather than proactive planning. This creates a broken system that fails both citizens and authorities.

Problem Placement

This problem affects two main groups: citizens who face civic issues daily and municipal departments who need to fix them. It covers multiple city departments like sanitation, roads, and streetlights. The problem exists in cities and towns across Pakistan where there is infrastructure but no proper system to report or track issues. This fits into the larger goal of using technology to make government services better, more transparent, and more accountable to the public.

Problem Relevance

Pakistan's cities are growing very fast and by 2030, more than 60% of people will live in urban areas. This puts extra pressure on infrastructure, which is already struggling. Every day, many civic problems go unreported or are not fixed because current complaint systems are old and not effective.

Poor infrastructure, like bad roads, costs the country money and affects millions of citizens who need safe and functional public spaces. Pakistan is moving towards smart cities, but there is still no proper system for citizens to report problems or for authorities to collect reliable data. Without this data, municipalities cannot find problem areas, see how serious issues are, or plan repairs well. This gap between smart-city plans and reality makes it hard to improve cities and manage them sustainably.

Problem Importance

For Citizens: Fixed roads and streetlights, updates on their complaints, and a voice in improving their neighborhoods.

For Municipalities: Better use of budget and staff, data to plan repairs, and greater public trust. For Urban Development: Smarter cities, better living conditions, and lasting infrastructure improvements across Pakistan.

Competitive Solutions Analysis

Local Competitors (Pakistan)

- Pakistan Citizen Portal (PCP): Nationwide coverage, connects departments.
  Weaknesses: slow responses, complaints can be closed without citizen approval, poor UI, no community verification, no data for planning.

- Municipal Helplines (LWMC 1139, WASA 1334, KMC 1339): Phone-based systems.
  Weaknesses: no tracking, low transparency, duplicate complaints, limited data, basic services only.

Regional Competitor (India)

- Swachhata-MoHUA: Works with local government. Weaknesses: limited features, manual verification, no AI.

International Competitors

- SeeClickFix (USA) & FixMyStreet (UK): Map-based reporting with voting.
  Weaknesses: no Urdu, strong internet needed, designed for more accountable governments.

2. Proposed Solution

Solution Overview

StreetLight is a centralized digital platform designed to improve communication between citizens and municipal authorities, along with the use of decentralized technology for immutability and temper proofing. It enables citizens to report civic issues quickly and allows authorities to manage, track, and resolve these issues in a transparent and efficient manner.

What We Are Developing:

- A citizen-friendly app to report issues with photos and location.
- Community verification where nearby residents confirm complaints before they are sent to authorities.
- A department dashboard for municipal officers to view verified complaints, update progress, and mark problems as resolved.
- An analytics engine that collects data over time to identify recurring problems and patterns across cities.

Nature of the Solution

StreetLight is a complete digital platform with three main components:

1. Citizen Reporting (Mobile App):

Citizens report problems using photos or videos. The app automatically captures their location using GPS. They add a description and submit with one tap. Each report gets a unique ID for real-time tracking.

2. Processing & Verification Engine:

The system automatically checks reports, identifies issue type and severity, finds duplicates, and allows nearby citizens to verify problems through voting. Reports are marked verified when enough people confirm them.

3. Municipal Dashboard (Web Panel):

City departments view verified reports, assign them to teams, and update status as work progresses. Officers attach before-and-after photos and use data for better planning and resource management.

How It Solves the Defined Problem

Identified Problem	Streetlight Solution
Multiple unorganized complaint channels	A single centralized digital platform
Slow and unclear response process	Real-time tracking and status updates
Duplicate or fake complaints	Community-based verification and automatic duplicate detection
Lack of transparency	Citizens can view progress and resolution proof
Poor planning due to lack of data	Analytics dashboard highlights trends and recurring issues

Core Features

Feature 1: Instant Issue Reporting

Citizens report problems using photos and location. This makes it simple for anyone to report issues quickly.

Feature 2: Automatic Problem Analysis

The system identifies what the problem is, how serious it is, and where it is located, helping authorities fix urgent issues first.

Feature 3: Community Verification

Nearby people confirm if complaints are real before they go to authorities, stopping fake reports and building trust.

Feature 4: Duplicate Detection

The system finds and combines repeated complaints about the same issue, reducing extra work and keeping data accurate.

Feature 5: Blockchain Security

Verified complaints are stored securely on blockchain so they cannot be changed or deleted, providing permanent proof.

Feature 6: Status Updates

Complaints move through stages (Pending → Verified → Assigned → In Progress → Resolved → Closed) and citizens get updates at each step.

Feature 7: Data Analytics Dashboard

Shows where problems happen most and how departments are performing, helping cities plan better.

Feature 8: Comments & Evidence Sharing

Citizens and authorities can add comments and photos to share information and work together on solutions.

Feature 9: CCTV Automation (Future Scope)

CCTV cameras can automatically detect civic issues without needing citizen reports, covering areas where fewer people report problems.

3. Functional Requirements

Sr. No.	Functional Requirement	Feature List
01	Citizens can report issues quickly	A mobile reporting service shall be provided for submitting issues with photos, GPS location, and instant report ID in under 2 minutes
02	Prevent fake and duplicate complaints	A community verification system shall be developed where nearby residents vote on complaints, requiring 3+ confirmations and ≥60% approval
03	Detect duplicate reports automatically	An automatic deduplication service shall be provided that identifies and merges duplicate reports within 30m radius and 14 days
04	Ensure complaint records cannot be changed	A blockchain integrity system shall be developed to store verified complaint hashes, preventing deletion or alteration without trace
05	Reports reach correct department automatically	An automated routing system shall be provided that sends verified complaints to responsible departments with immediate notifications

06	Citizens track complaint status	A real-time tracking service shall be developed showing progress through all stages with notifications at each step
07	Departments remain accountable for resolutions	A proof of completion system shall be provided requiring officers to upload before/after photos and work orders before closure
08	Identify problem areas automatically	An analytics dashboard shall be developed with interactive maps showing complaint density, issue types, and severity levels
09	Measure municipal performance	A performance tracking system shall be provided monitoring resolution time, completion rates, and deadline adherence per department
10	Citizens can add updates and evidence	A commenting service shall be developed for posting follow-up comments and photos with automated content moderation
11	Authorities detect issues using AI	An AI analysis system shall be provided for image classification, severity assessment, and attribute extraction for prioritization
12	Data exportable for urban planning	A data export service shall be developed allowing admins to download complaint data in CSV/GeoJSON formats

13	Prevent officer manipulation	An audit logging system shall be provided tracking all officer actions with timestamps and user IDs
14	Web access for municipal departments	A web dashboard shall be developed for officers to view complaints, update status, and manage workflow
15	Departments see assigned issues	A department-specific view shall be provided displaying complaints with issue details, location, and assigned officer
16	Admins manage users and departments	An admin panel shall be developed for user management, role assignment, and system activity monitoring
17	Maintain complaint history	A permanent storage system shall be provided for all complaint data, updates, actions, and feedback for audits and planning

User Roles & Responsibilities

No.	User Role	Responsibilities
01	Citizen / Reporter	◆ Register and create an account on the mobile app ◆ Report civic issues using photos, videos, and GPS location ◆ Verify nearby complaints from other citizens ◆ Track complaint status in real time ◆ Accept or reject complaint resolutions ◆ Add comments and upload follow-up evidence ◆ Receive notifications and status updates ◆ View municipal response metrics
02	Municipal Officer	◆ Access assigned complaints through the web dashboard ◆ Update complaint status at each workflow stage ◆ Upload before and after photos as proof of work ◆ Record work details and expected completion dates ◆ Respond to citizen comments ◆ Monitor assigned complaints and deadlines
03	Department Head	◆ View all departmental complaints ◆ Assign complaints to officers or teams ◆ Monitor officer performance and resolution times ◆ Review escalated complaints ◆ Access analytics and hotspot maps ◆ Export complaint data for planning ◆ Manage user roles and permissions ◆ Review system audit logs
04	City Planner / Analyst	◆ Access analytics dashboards and maps ◆ Analyze complaint trends and hotspot areas ◆ Review historical complaint patterns ◆ Export reports for urban planning and budgeting

05	System Administrator	◆ Manage user accounts and permissions ◆ Configure municipal departments ◆ Monitor system activity and audittrails ◆ Handle system maintenance and technical issues ◆ Export complaint data for external analysis

4. Non - Functional Requirements

i. Development Environment

- **Agile Process:**

➢ StreetLight will be developed in small, testable iterations. Core features like report submission, detection, and tracking are built first, followed by analytics and future CCTV integration.

- **Programming Tools:**

➢ ***Mobile:*** Flutter SDK (iOS & Android)

➢ ***Web:*** React.js with Node.js

➢ ***Backend:*** Python (FastAPI) for APIs and AI services

- **Build & Dependency Management:**

➢ Flutter build system, npm/yarn for JavaScript, pip with virtual environments for Python

- **Containerization:**

➢ Docker for backend and AI services

➢ Docker Compose for multi-service local development

- **Automation:**

➢ GitHub Actions for CI/CD: linting, build validation, unit tests

➢ Failed builds block merges to ensure code quality

Ensures consistent builds, rapid development, and easy onboarding for developers.

ii. Deployment Details

- **Cloud Hosting:**
  - Free/low-cost cloud platforms; no physical servers required.

- **Mobile App:**
  - Android via Google Play Store
  - iOS via TestFlight
  - All communication over HTTPS

- **Web Dashboard:**
  - Hosted on Vercel/Netlify (free tier)
  - Responsive design for all screen sizes

- **Backend Services:**
  - APIs on Render/Railway/Fly.io (free tier)
  - Stateless services enabling horizontal scaling
  - Zero-downtime deployments using rolling restarts

- **Database & Storage:**
  - PostgreSQL with PostGIS
  - Cloud storage for images/videos with signed URLs
  - Daily automated backups

Deployment ensures accessibility, reliability, and low operational cost for municipal budgets.

iii. Archiving & Version Control

- **Version Control:**
  - Git (GitHub free tier)

- **Branching Strategy:**
  - main: Stable production releases
  - develop: Integration branch for active development
  - feature/: Individual developer work merged via pull requests

- **Archiving & Releases:**
  - Semantic versioning (v1.0.0, v1.1.0…)
  - Release notes for features, bug fixes, and changes
  - Database migration scripts version-controlled
  - Full audit trail with timestamps and developer attribution

Ensures transparency, rollback ability, and long-term maintainability.

iv. Security & DevSecOps Practices

- **Secure Communication:** HTTPS with TLS 1.2+, automated SSL via Let's Encrypt

- **Authentication & Authorization:**
  - Firebase Authentication
  - Role-based access (citizens, officers, admins)
  - JWT tokens for session management

- **Secrets Management:** API keys and credentials stored in environment variables; no hardcoding

- **Data Protection:**
  - AES-256 encrypted storage
  - Automatic face/license plate blurring
  - Minimal PII retention; location data aggregated

- **Code Security:**
  - Automated linting (ESLint, Pylint), static analysis (SonarQube)
  - Mandatory code reviews

- **Operational Security:**
- Access logs, rate limiting, DoS protection
- Monitoring for suspicious activity

Ensures StreetLight is secure, privacy-preserving, and compliant with modern software standards.

5. Technology Stack

System Architecture Overview

Component	Description
Central Backend	Fast processing, analytics & workflow management
Blockchain Layer	Complaint immutability & transparency
Core Components	Mobile App, Web Dashboard, Python Backend, Database, Blockchain

Frontend Technologies – Mobile Application (Citizen Interface)

Category	Technology
Framework	Flutter (Dart)
State Management	Provider
API Communication	Dio
Local Storage	Hive
Maps & Geolocation	OpenStreetMap + GPS
Media Handling	Image Picker, Flutter Compress
Push Notifications	Firebase Cloud Messaging (FCM)
Localization	Urdu / English

Frontend Technologies – Web Dashboard (Municipal & Admin Interface)

Category	Technology
Framework	React.js
State Management	Redux Toolkit
HTTP Client	Axios
UI Library	Material UI
Data Visualization	Recharts
Maps	Leaflet.js
Styling	Tailwind CSS
Build Tool	Vite
Hosting	Vercel

Backend Technologies

Category	Technology
Language	Python
Framework	FastAPI
Authentication	JWT
ORM	SQLAlchemy
Background Jobs	Celery + Redis
Logging & Monitoring	Structured Logging
AI / Image Processing	PyTorch, OpenCV

Blockchain Layer

Category	Technology
Blockchain Network	Ethereum / Polygon (Testnet)
Smart Contracts	Solidity
Usage	Store complaint hashes for immutability

Data Management & Storage

Category	Technology
Primary Database	PostgreSQL
Geospatial Support	PostGIS
Caching	Redis
Media Storage	Firebase Storage

Notification System

Category	Technology
Push Notifications	Firebase Cloud Messaging (FCM)

Development & Deployment Tools

Category	Tools
Version Control	GitHub
CI/CD	GitHub Actions
Containerization	Docker
Testing	Pytest, Flutter Test
Diagrams	Draw.io
API Documentation	Swagger UI
UI/UX Design	Figma

6. Conceptual Design

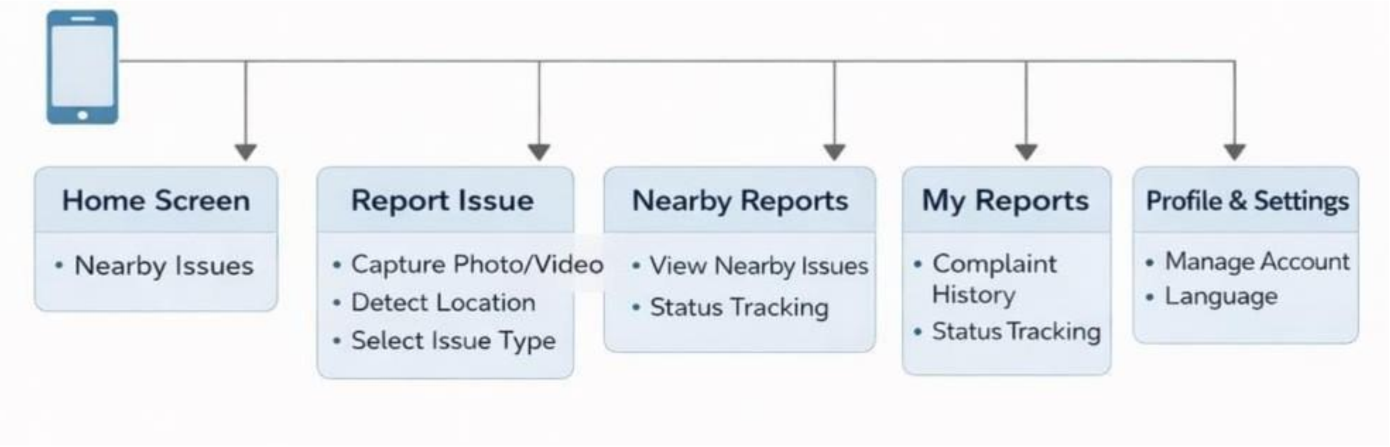
Conceptual System Architecture

- Multi-layered system: Citizen Interaction → Processing & Verification → Municipal Management & Planning
- Citizen-focused, transparent, accountable
- Data-driven insights for urban planning

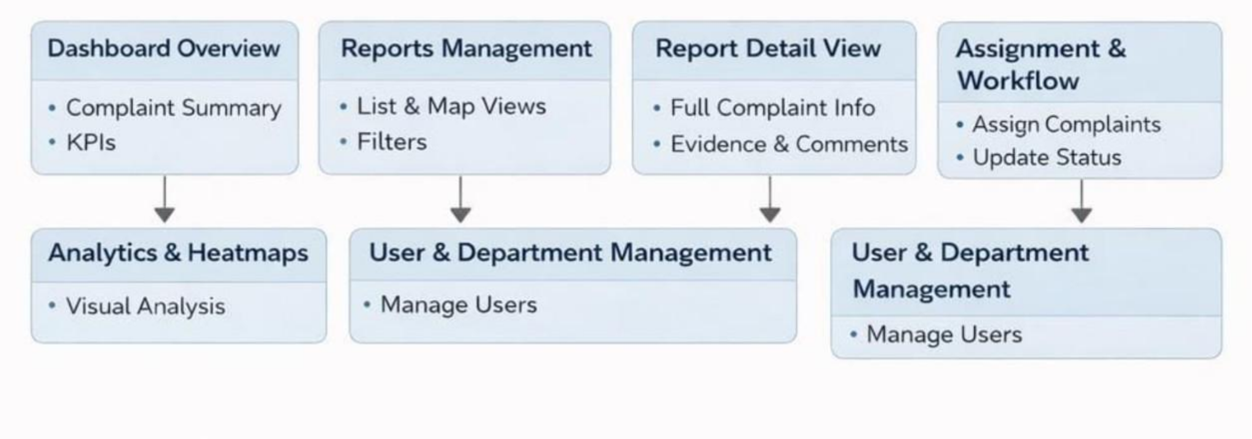
Conceptual Site Map

This site map shows main functional areas for different users in the system

Mobile App



Municipal Dashboard



Conceptual Navigation Flow

- **Citizen Flow:**

Login → View nearby issues → Submit new issue → Receive report ID → Community verification → Track status → Review resolution

- **Municipal Flow:**

Officer login → View assigned complaints → Review → Assign → Update status → Upload before/after photos → Mark resolved → Notify citizen

Core Conceptual Entities

- User: Citizens, officers, admins, planners, heads
- Civic Issue Report: Issue category, location, evidence, submission time, status
- Community Verification: Votes by nearby citizens
- Work Order: Municipal actions linking report to officers
- Department: Responsible for issue type; automated routing

- **Complaint Lifecycle**

1. Pending → 2. Verified → 3. Assigned → 4. In Progress → 5. Resolved → 6. Closed

Verified complaints treated as permanent records.

- **Accountability & Transparency Model**
  - Assign officers, deadlines, require photo proof
  - Citizen approval/escalation
  - Automatic escalation if deadlines missed

7. Special Features List

1.Hardware: CCTV integration, mobile GPS & camera

2.Theoretical: AI/ML for severity, duplicate detection, predictive analytics

3.Programmatic: Blockchain, APIs, Flutter, React.js + Redux

4.Techniques: Community verification, smart deduplication