Project Design Phase-II

# Technology Stack (Architecture & Stack)

|  |  |
| --- | --- |
| Date | 10 June 2025 |
| Team ID | LTVIP2025TMID59973 |
| Project Name | Citizen AI – Intelligent Citizen Engagement Platform |
| Maximum Marks | 4 Marks |

## Technical Architecture:

The deliverable includes an architectural diagram showing how citizens interact with the system through web and mobile interfaces. Reported issues are processed by an AI module (for classification and prioritization), stored securely in a cloud database, and routed to appropriate government departments via secure APIs.  
  
Guidelines Addressed:  
- Processes (AI classification, reporting, feedback, escalation)  
- Infrastructure: Mix of Cloud + Web  
- External APIs: e.g., Aadhar for identity, GIS APIs  
- Storage: Cloud databases and file storage  
- Interface to AI/ML models (e.g., for classification, NLP)

### Table-1: Components & Technologies

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology |
| 1 | User Interface | Mobile App and Web UI for citizens and officials | Flutter (Mobile), React.js (Web), Tailwind CSS |
| 2 | Application Logic-1 | Issue Reporting, Tracking, Feedback Modules | Node.js, Express.js |
| 3 | Application Logic-2 | AI-based Issue Categorization and Prioritization | Python, Scikit-learn, TensorFlow |
| 4 | Application Logic-3 | Natural Language Processing & Chat Assistant | OpenAI API / Rasa / IBM Watson Assistant |
| 5 | Database | Structured storage for user data and issue logs | PostgreSQL |
| 6 | Cloud Database | Cloud-hosted database for scalability and access | AWS RDS / Google Cloud SQL |
| 7 | File Storage | Media files from citizen reports | AWS S3 / Google Cloud Storage |
| 8 | External API-1 | Location services, Maps integration | Google Maps API |
| 9 | External API-2 | Citizen identity verification | Aadhar API |
| 10 | Machine Learning Model | Classifies issue types and recommends actions | Custom model on TensorFlow / Scikit-learn |
| 11 | Infrastructure | Deployed on Cloud and scalable container environment | Kubernetes on AWS EKS / GCP GKE |

### Table-2: Application Characteristics

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology |
| 1 | Open-Source Frameworks | Frontend and backend frameworks | React, Node.js, Rasa, PostgreSQL |
| 2 | Security Implementations | Auth, encryption, IAM, API gateway security | JWT, HTTPS, OAuth 2.0, AWS IAM |
| 3 | Scalable Architecture | Microservices and containerized deployment | Docker, Kubernetes |
| 4 | Availability | Load balancing and multi-zone cloud deployment | AWS ALB/ELB, Auto Scaling |
| 5 | Performance | API caching, async queues, optimized DB access | Redis, CDN, NGINX, PostgreSQL indexing |