**Project Design Phase-II**
**Technology Stack (Architecture & Stack)**

| | |
|---|---|
| Date | 27 October 2022 |
| Team ID | Team-593022 |
| Project Name | Project - Vehicle Counter |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

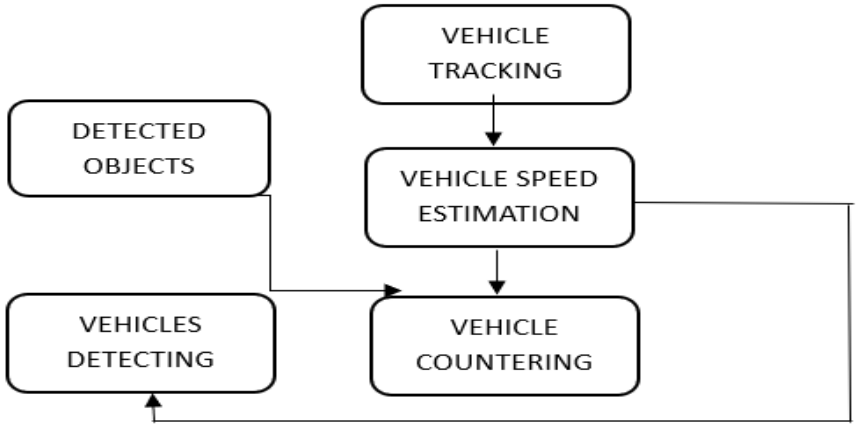

**Table-1 : Components & Technologies:**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | How user interacts with application | Web UI |
| 2. | Application Logic-1 | Logic for vehicle countering application | Python |
| 3. | Application Logic-2 | Logic for sorting and managing vehicle data | MySQL |
| 4. | Application Logic-3 | Logic for providing interface | IBM Watson Assistant |
| 5. | Database | Data Type, Configurations etc. | MySQL |
| 6. | Cloud Database | Database Service on Cloud | IBM Cloudant |
| 7. | File Storage | File storage requirements | IBM Block Storage |
| 8. | External API-1 | Purpose of External API used in the application | IBM Object recognition model |
| 9. | External API-2 | Purpose of External API used in the application | IBM Assistant |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Vehicle countering Model - KNN Classifier |
| 11. | Infrastructure (Server / Cloud) | Application Deployment on Local System | IBM Cloud factory |

**Table-2: Application Characteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | List the open-source frameworks used | TensorFlow |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Authentication and authorization, data encryption, firewalls |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Three-tier architecture, with the application tier horizontally scalable and the database tier vertically scalable |

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Application tier deployed behind a load balancer, database replicated to multiple zones |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Use of a high-performance programming language (Python), use of an optimized machine learning library (TensorFlow), use of a high-performance database (IBM Cloudant NoSQL Database) |

**References:**

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d