**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 28/10/23 |
| Team ID |  |
| Project Name | Graphical Advantages: A tableau Exploration  of Top Manga |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

|  |  |  |
| --- | --- | --- |
| User Interface | Integration | Backend |
| The UI shows the output to the user.  User passes image. | Integration of the user interface (HTML, CSS, or Bootstrap)  Development of the model (CNN or Transfer Learning) | Importing the saved model  Model training using image augmentation.  Saving the model  Creation of a Flask app |

**Table 1- Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Component | Description | Technology |
| 1 | User Interface | How user interacts with application e.g. Web UI | HTML, CSS, JavaScript / React Js/ Angular Js etc. |
| 2 | Application logic | Logic for a process in the application | Tabeleu/IBM Cognos Analytics |
| 3 | Database | Manga title, author, sales data, and reader preferences | MySQL, NoSQL, etc. |
| 4 | File Storage/ Data | File storage requirements for Storing the dataset | Local System, Google Drive Etc |
| 5 | Recommendation System | Algorithm for suggesting manga based on user preferences | Machine Learning, Content-Based Filtering |
| 6 | Digital Publishing | Platform for digital manga distribution | Amazon Kindle, ComiXology, Manga Rock, etc. |
| 7 | Community Features | Forums, comments, and social features for reader engagement | PHP, MySQL, Social Media Integration |
| 8 | Cloud Hosting | Infrastructure for web and data hosting | Amazon Web Services, Google Cloud Platform, etc. |
| 9 | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud  Local Server Configuration:  Cloud Server Configuration : | Local, Cloud Foundry, Kubernetes, etc. |

**Table 2-Application Characteristics**

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Characteristics | Description | Technology |
| 1 | Open-Source Frameworks | List the open-source frameworks used | Django, Angular Js, Spark |
| 2 | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | SSL/TLS, Authentication |
| 3 | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro services) | Microservices, Load Balancers |
| 4 | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Cloud Hosting |
| 5 | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | CDNs, Caching |