|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Component | Description | Technologies used |
| 1 | User Interface | How user interacts with the application | HTML, CSS, JavaScript / Angular Js / React Js, etc |
| 2 | Text Data Collection | Gathering airline reviews data | Web Scraping, API (e.g., social media platforms) |
| 3 | Data preprocessing | Cleaning and preprocessing of raw data | Natural Language Processing (NLP) tools (e.g., NLTK, SpaCy) |
| 4 | Feature Extraction | Extracting features from text data | Word Embeddings (e.g., Word2Vec, GloVe) |
| 5 | Machine Learning Model | Classification algorithm for reviews | Scikit-learn, TensorFlow, PyTorch |
| 6 | Model Evaluation | Assessing model performance using metrics | Scikit-learn |
| 7 | Web Application | Interface for users to interact with the model | Flask, Django |
| 8 | Containerisation | Packaging application and dependencies into containers | Docker |
| 9 | Cloud Platform | Deployment on a cloud platform for scalability | AWS, Azure, Google Cloud |
| 10 | Monitoring and Logging | Monitoring application behavior and logging | Prometheus, Grafana |
| 11 | CI/CD Pipeline | Automating testing and deployment processes | Git, Jenkins, GitLab CI |
| 12 | Security | Implementing authentication and data encryption | SSL/TLS, Secure APIs                | |

**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

**Table1:Components and Technologies**

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| **1** | Open-Source Frameworks | List the open-source frameworks used | Scikit-learn, TensorFlow, PyTorch, NLTK, SpaCy |
| **2** | Security Implementations | List all the security/access controls implemented, use of firewalls, etc. | Encryption(e.g.,SSL/TLS),IAM Controls,OWASP,SHA-256, Secure APIs |
| **3** | Scalable Architecture | Justify the scalability of architecture (3-tier, Micro-services) | Microservices architecture, Docker, Kubernetes |
| **4** | Availability | ustify the availability of the application (e.g., use of load balancers, distributed servers | LoadBalancers, Redundancy, Failover Mechanisms |
| **5** | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s | Caching Strategies, Content Delivery Networks (CDN), LoadTesting, Performance Monitoring |