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

Date	27 October 2023
Team ID	Team-593093
Project Name	Project – Eye Disease Prediction Using Deep Learning Project
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

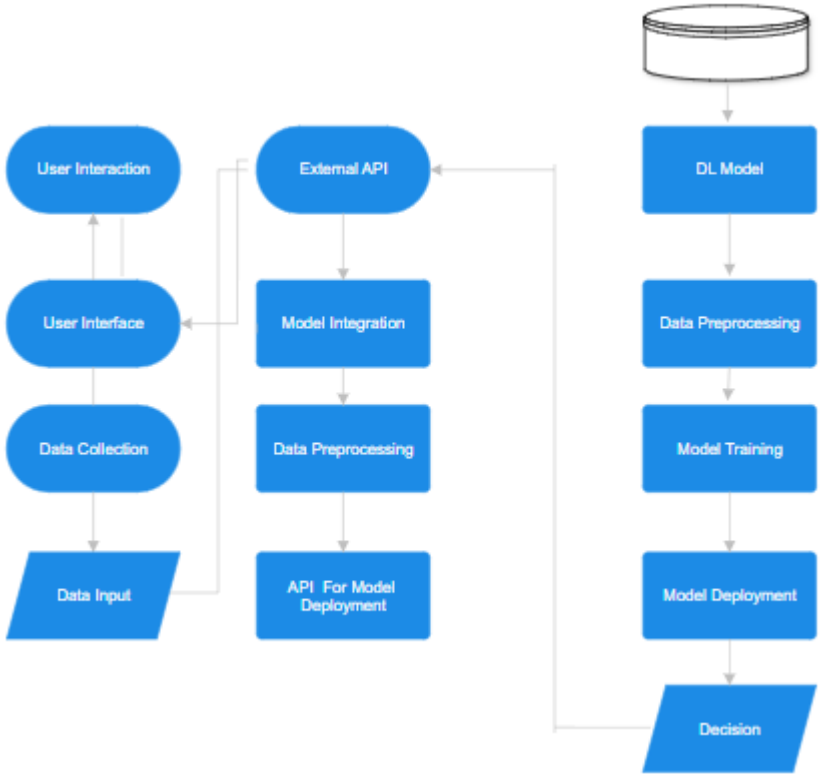

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Interface for user interaction the application along with creating an user friendly interface.	Web-based UI(HTML, CSS, JavaScript / Angular Js/React Js).
2.	Application Logic-1	Core Logic responsible for handling user requests.	Python, Flask, FastAPI or Node.js
3.	Data Collection	Gathering eye disease images data from hospital database.	Web Scraping, Data warehouses, ETL tools
4.	Data Input	Handling and preprocessing user provided input.	Forms, API's or command line input
5.	External API	Integration with external data sources or services.	RESTful API's (
6.	Cloud Database	Database Service on Cloud (Storage and management of structured data).	Amazon RDS, Google Cloud SQL, Azure SQL.
7.	Model Integration	Interface for integrating deep learning model.	RESTful API endpoints(Flask, FastAPI, JSON,XML)
8.	API Model Deployment	Responsible for deploying deep learning models as APIs, enabling real-time predictions and external interaction. It ensures model accessibility and scalability.	Docker, Kubernetes etc.
9.	Deep Learning Model	The predictive model for eye diseases using eye images.	Scikit-Learn, TensorFlow, PyTorch, Transfer Learning Techniques like VGG, Inception etc.
10.	Data Preprocessing	Data preparation here, augmentation, normalization, resizing of images etc.	Pandas, Numpy , Scikit-learn or custom scripts.
11.	Model Deployment	Hosting and Serving the deep learning model.	Streamlit, Flask, FastAPI.
11.	Infrastructure (Server / Cloud)	Underlying Cloud infrastructure and resources.	AWS, Google Cloud, Azure or on-premises servers like local,Cloud Foundry etc. .

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizing open-source frameworks for model development and deployment, ensuring cost-efficiency and flexibility which makes easy for the user to predict the disease.	Scikit-Learn, TensorFlow, PyTorch for model development. – Streamlit orFlask or FastAPI for API deployment. - Kubernetes for container orchestration. - Jupyter Notebook for model prototyping and development.
2.	Security Implementations	Implementing robust security measures to protect sensitive data, model APIs, and ensure user data privacy which helps hospitals to protect the patient's data and hospital's data from the hackers.	OAuth 2.0 or JWT for user authentication. Encryption (HTTPS/SSL) for data in transit. Role-based access control. Regular security audits and updates. Compliance with industry standards (e.g., GDPR).
3.	Scalable Architecture	Designing a scalable architecture that can handle growing data volumes and user demands which can manage the huge inflow of user demands assuming as a big data.	- Microservices architecture for modularity and scalability. - Containerization with Docker and orchestration with Kubernetes. - Load balancers for distributing traffic. - Auto-scaling based on resource usage.
4.	Availability	Ensuring high availability and minimal downtime for the application to support continuous data analysis and prediction which helps in predicting eye diseases accurately and rapidly.	Redundancy in database and API deployment. - Geographically distributed data centers or cloud regions. - Monitoring and alerting systems (e.g., Prometheus, Grafana). - Failover mechanisms for fault tolerance
5.	Performance	Optimizing application performance to provide quick insights and predictions	Caching mechanisms for frequently accessed data. - Model optimization (e.g., quantization) for faster inference. - Load testing and performance tuning

6.	User-Friendly Interface	Creating an intuitive and user friendly interface for data input, visualization, and interact.	HTML, CSS, JavaScript for web-based UI. - React or similar frameworks for responsive design. - Data visualization libraries (e.g., D3.js). - User experience (UX) testing and design principles
7.	Interoperability and Accuracy	Ensuring seamless integration with external systems and maintaining high prediction accuracy.	- RESTful API design for interoperability. - Integration with external data sources (e.g., weather data). - Continuous model monitoring and retraining for accuracy improvement. - Data preprocessing techniques to enhance model accuracy