

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

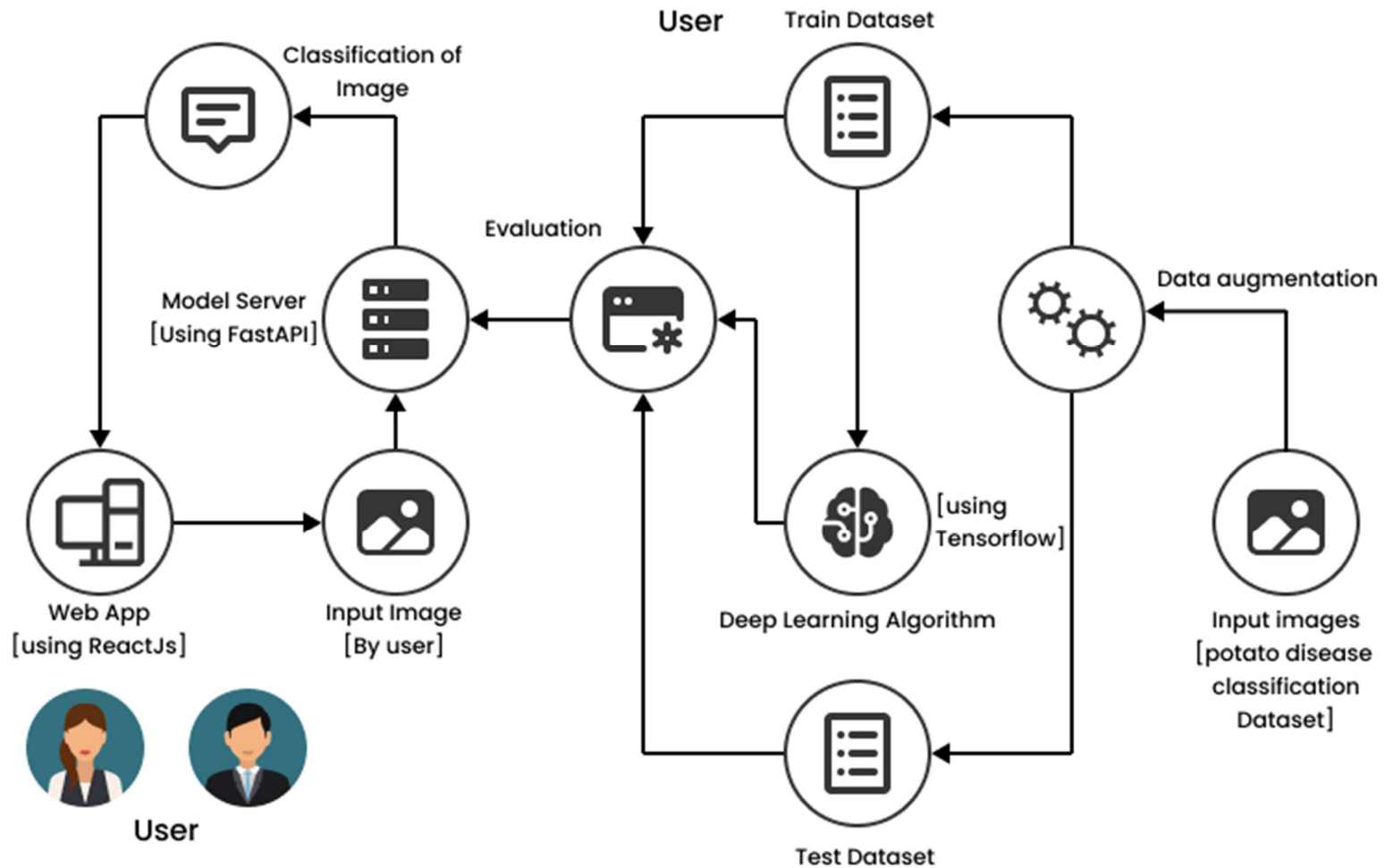
|               |                               |
|---------------|-------------------------------|
| Date          | 18 November 2023              |
| Team ID       | Team - 591844                 |
| Project Name  | Potato Disease Classification |
| Maximum Marks | 4 Marks                       |

**Technical Architecture:**

Technical architecture encompasses the fundamental structure and design of a software system, outlining its key components, their interactions, and the underlying framework that enables its functionality. This includes hardware infrastructure, software modules, databases, networking protocols, and interfaces. It also addresses critical considerations such as data storage, scalability, security measures, and compliance with industry standards. The architecture dictates how the system handles increasing loads and ensures optimal performance. It encompasses deployment strategies, whether on-premises or on cloud platforms, as well as procedures for error handling, recovery, and backups. Integration with external services and APIs is also specified, enabling seamless interaction with third-party applications. Furthermore, it delineates development, testing, staging, and production environments, ensuring consistency across different stages of the software development life cycle. Monitoring and logging mechanisms are put in place to track system behavior, performance metrics, and errors. Maintenance procedures and strategies for upgrades are defined to keep the system up-to-date and efficient. Documentation is crucial, providing insights into architectural decisions and best practices, facilitating knowledge transfer among team members and aiding in troubleshooting. In essence, technical architecture serves as the foundational blueprint for the construction and maintenance of a robust and effective software system.

The Deliverable shall include the architectural diagram as below and the information as per the Table1 & Table 2

### Architectural Diagram



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

| S.No | Component              | Description   | Technology  |
|------|------------------------|---|-------------|
| 1.   | User Interface Design  | To select and display the image for which the user wants to find out the health status of potato leaf.  | ReactJs     |
| 2.   | User Experience        | Image classification of potato leaf disease detection delivers quick response.  | ReactJs     |
| 3.   | Application Logic - 1  | Image Processing: Preprocess the user-uploaded image.   | TensorFlow. |
| 4.   | Application Logic - 2  | Feature Extraction and Model Prediction: Extract features using a pre-trained model. Predict leaf disease using a sequence-to-sequence model. | TensorFlow. |
| 5.   | Application Logic - 3  | Display the health status of the leaf to the user.  | ReactJs     |
| 6.   | Machine Learning Model | Image classification DL models automatically describe images classes trained on.  | TensorFlow. |
| 7.   | FastAPI Integration    | Used to host the ml model so that it can be accessed using api.   | FastAPI     |

**Table-2**  
**Application Characteristics:**

| S.No | Characteristics             | Description  | Technology                   |
|------|-----------------------------|--|------------------------------|
| 1.   | Open-Source Frameworks      | List the open-source frameworks used   | TensorFlow, VS Code, FastAPI |
| 2.   | Availability                | Achieve high availability for image classification with load balancing, redundancy.                                  | Tensorflow, FastAPI          |
| 3.   | Performance                 | Image classification performance is assessed through metrics like accuracy, model architecture, and inference speed. | TensorFlow.                  |
| 4.   | Image Upload and Processing | Allow users to upload images for classification with backend processing.   | FastAPI                      |
| 5.   | Responsive Web Design       | Ensure the web application is accessible and user-friendly on various devices.                                       | ReactJs                      |