Project Design Phase-I
Solution Architecture

Date	10 November 2023
Team ID	591844
Project Name	Project – Image Caption Generator
Maximum Marks	4 Marks

Solution Architecture:

The web application architecture consists of a frontend written in ReactJS, a Python backend using FastAPI, and a machine learning model for disease classification.

Frontend:

The ReactJS frontend allows users to upload images of potato plant leaves and displays the disease classification results. It has components for:

- Home page
- Image upload form
- Loading state
- Results display
- Navigation bar

It retrieves predictions from and sends images to the FastAPI backend.

Backend:

The FastAPI server provides the REST API endpoints for image classification. It handles:

- Image uploads
- Saving images
- Sending images to ML model
- Returning predictions
- User authentication

It connects to a SQLite database for saving images and a Redis cache for improving performance.

Machine Learning Model:

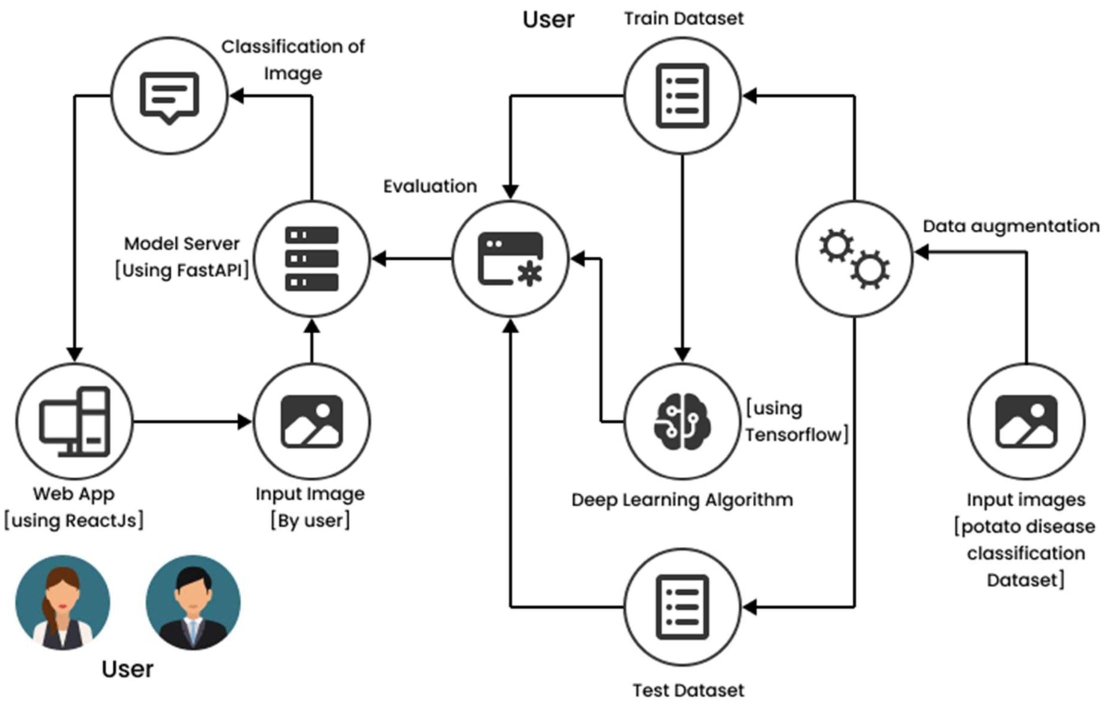
A convolutional neural network (CNN) like ResNet50 is used for image classification. It takes images as input and outputs a predicted disease class and confidence percentage.

The model is trained on a dataset of potato plant images labeled with diseases. It is exported and served using TensorFlow Serving for low-latency predictions.

Deployment:

The frontend is hosted on Vercel and the FastAPI backend server is containerized with Docker and deployed on Render. The TF Serving model server is hosted on a GPU instance.

Solution Architecture Diagram (for Potato Disease Classification) :



User
Train Dataset
Classification of Image
Evaluation
Model Server [Using FastAPI]
Data augmentation
[using Tensorflow]
Deep Learning Algorithm
Web App [using ReactJs]
Input Image [By user]
Input images [potato disease classification Dataset]
User
Test Dataset