****

**COMSATS University Islamabad (CUI) Attock Campus**

Software Requirement Specification  
(SRS DOCUMENT)

for

**Wheat Shield**

Version 1.0

***By***

**Kashif Hussain CIIT/FA20-BCS-019/ATK**

**Muhammad Zubair CIIT/FA20-BCS-041/ATK**

***Supervisor*Mr. Muhammad Wasim Khan**

*Bachelor of Science in Computer Science (2020-2024)*

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Application Evaluation History

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

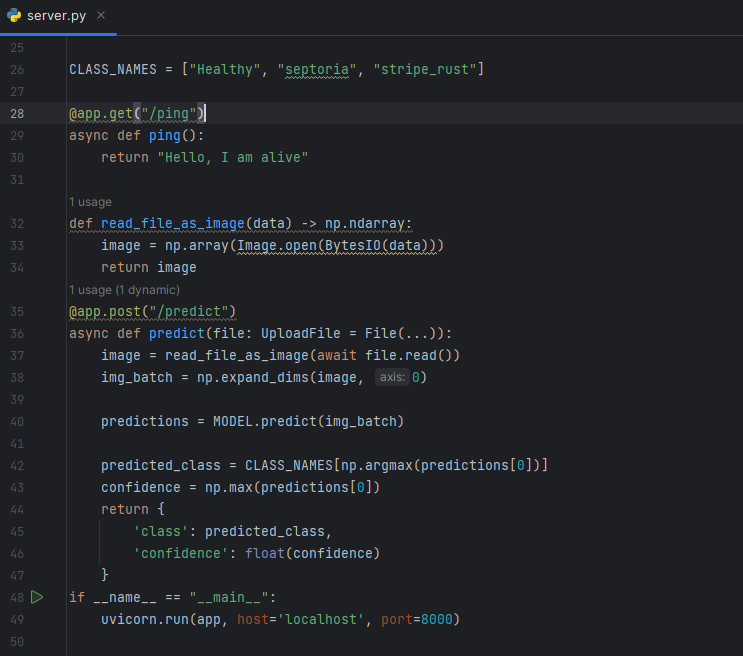
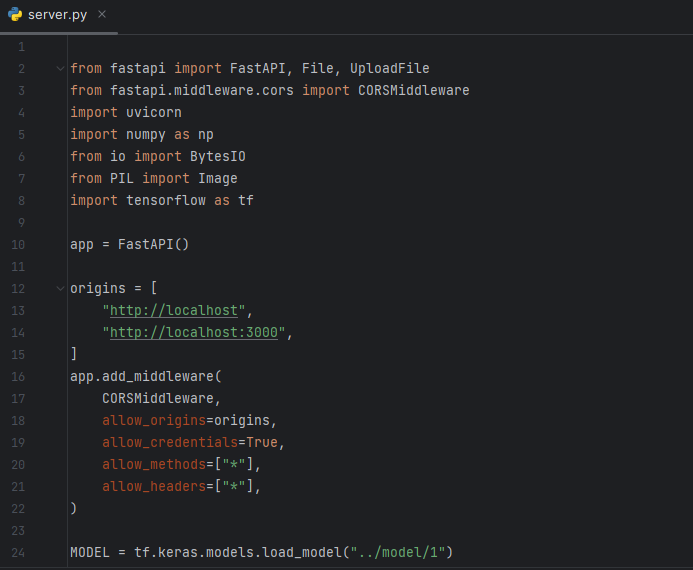
Supervised by

Mr. Muhammad Wasim Khan

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**API Explanation :**

Code :



**Detailed Explanation:**

1. **Importing Libraries:**
   * Libraries such as FastAPI, NumPy, BytesIO, PIL (Pillow), and TensorFlow are imported.
   * FastAPI is a Python web framework for building APIs quickly and efficiently.
2. **FastAPI Setup:**
   * FastAPI instance is created with **FastAPI()**.
   * CORS (Cross-Origin Resource Sharing) middleware is configured to allow specific origins for API requests.
   * Origins are set to "[http://localhost](http://localhost/)" and "[http://localhost:3000](http://localhost:3000/)" to handle cross-origin requests.
3. **Loading Pre-trained Model:**
   * The pre-trained neural network model is loaded using TensorFlow's **tf.keras.models.load\_model** method from the specified path.
   * The loaded model is used for making predictions.
4. **Class Names Initialization:**
   * **CLASS\_NAMES** list contains the class labels for the model: "Healthy", "septoria", and "stripe\_rust".
   * These labels are used to interpret the predictions made by the model.
5. **Ping Endpoint:**
   * An endpoint at **/ping** responds with "Hello, I am alive" to indicate that the API is running and accessible.
   * This endpoint is primarily used for basic health checks and server availability.
6. **Image Processing Function:**
   * **read\_file\_as\_image** function takes image data as input and converts it into a NumPy array using Pillow's Image module.
   * The function reads the image from the byte data provided by the user during the API call.
7. **Prediction Endpoint:**
   * The main prediction endpoint is defined at **/predict** and accepts an uploaded file as input.
   * The uploaded file is processed into an image array and expanded to create a batch of size 1.
   * The model predicts the class probabilities for the input image batch.
   * The predicted class is determined by finding the index with the maximum probability and mapping it to the corresponding class name.
   * The confidence score is obtained by finding the maximum probability in the predictions.
   * The endpoint returns a JSON response containing the predicted class and confidence score.
8. **Server Launch:**
   * The FastAPI application is run using **uvicorn.run** on "localhost" at port 8000.
   * The API becomes accessible for handling requests at the specified endpoint URLs.

**Summary:**

1. **Setup:**
   * FastAPI framework is utilized to create a robust API for image classification.
   * CORS middleware is configured to handle cross-origin requests from specific origins.
2. **Model Integration:**
   * A pre-trained neural network model is loaded, enabling the API to make accurate predictions.
3. **Endpoints:**
   * **/ping** endpoint provides a basic health check response.
   * **/predict** endpoint processes uploaded images, predicts classes, and returns results.
4. **Image Processing:**
   * Images uploaded to the API are processed using Pillow, converting them into suitable formats for the model.
5. **Response Format:**
   * Predictions are returned as JSON responses, including predicted class and confidence score.
6. **Server Launch:**
   * The API is launched locally on "localhost" at port 8000, making it accessible for incoming requests.