**📘-GrainPalette: A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning**

**Project Documentation**

**🔖 1. Introduction**

**Project Title:**  
GrainPalette – A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning

**Team ID:** LTVIP2025TMID42259  
**Team Leader:** SAMA VENKATA SAI KARTHIK

**Team Members: EDIGA VENKATADRI**

**Team Members: ANNAPUREDDY HEMA LATHA**

**Team Members: IRAGAM REDDY GOWRI SREEJA**

**Overview:**  
GrainPalette is a deep learning-based web application for classifying rice varieties from images. It uses transfer learning with CNNs to accurately identify rice types, helping farmers, millers, and exporters ensure rice quality. The system includes an intuitive web interface for uploading rice image and getting instant predictions.

**📌 2. Project Overview**

**Purpose:-**  
To automate rice variety identification using image-based deep learning, reducing manual errors in rice grading.

**Key Features:-**

· Upload rice images for instant classification

· High-accuracy model based on ResNet50 transfer learning

· Web app with user-friendly interface

· Displays confidence score for predictions

· Expandable to include rice disease detection

**🏗️ 3. System Architecture**

Frontend (HTML/CSS/JS + Flask/Jinja):

· HTML/CSS for static pages

· Jinja2 templates for dynamic pages (login, home, about, upload)

· JavaScript for image upload handling

**Backend (Flask + PyTorch):**

· REST endpoints for uploading and predicting rice images

· ResNet50 model fine-tuned on rice dataset

· Numpy/Pandas for preprocessing

· Pillow/OpenCV for image transformations

Database (Optional / Future):

· SQLite/MySQL for saving prediction history or user accounts

· Not required for initial phase

**⚙️ 4. Setup Instructions:**

**Prerequisites:**

· Python 3.9+

· pip

· virtualenv (recommended)

Installation & Setup:

# Clone repository

git clone https://github.com/your-username/grainpalette.git

# Setup virtual environment

cd grainpalette

python -m venv venv

venv\Scripts\activate # Windows

source venv/bin/activate # Linux/Mac

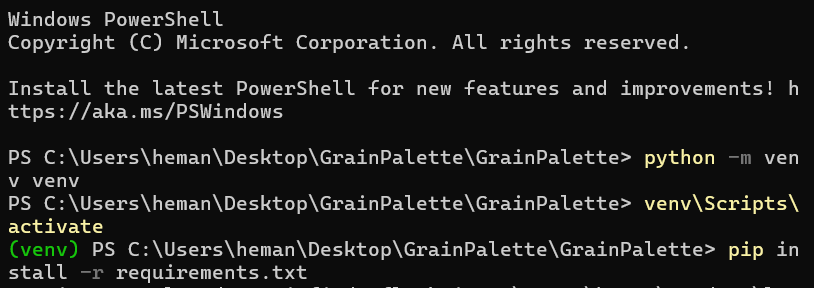
# Install dependencies

pip install -r requirements.txt

# Run the Flask server

python app.py

Frontend & Backend Terminals:-



Screenshot 2025-06-28 125735

Screenshot 2025-06-28 125752Screenshot 2025-06-28 125802

**📁 5. Folder Structure :-**

**grainpalette/**

**├── app.py**

**├── /templates/**

**│ ├── login.html**

**│ ├── home.html**

**│ └── ...**

**├── /static/**

**│ ├── css/**

**│ ├── js/**

**│ └── images/**

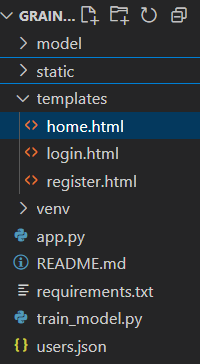
**├── /model/**

**│ └── rice\_classifier.pth**

**├── /utils/**

**│ └── preprocessing.py**

**└── requirements.txt**

****

**▶️ 6. Running the Application**

**# Start the Flask server**

**python app.py**

**📡 7. API Documentation**

|  |  |  |
| --- | --- | --- |
| **METHOD** | **ENDPOINT** | **DESCRIPTION** |
| **POST** | **/PREDICT** | **UPLOAD RICE IMAGE AND GET CLASSIFICATION** |
| **GET** | **/** | **HOME PAGE** |
| **GET** | **/ABOUT** | **ABOUT PROJECT** |
| **GET** | **/CONTACT** | **CONTACT US PAGE** |

**🔐 8. Authentication & Authorization :-**

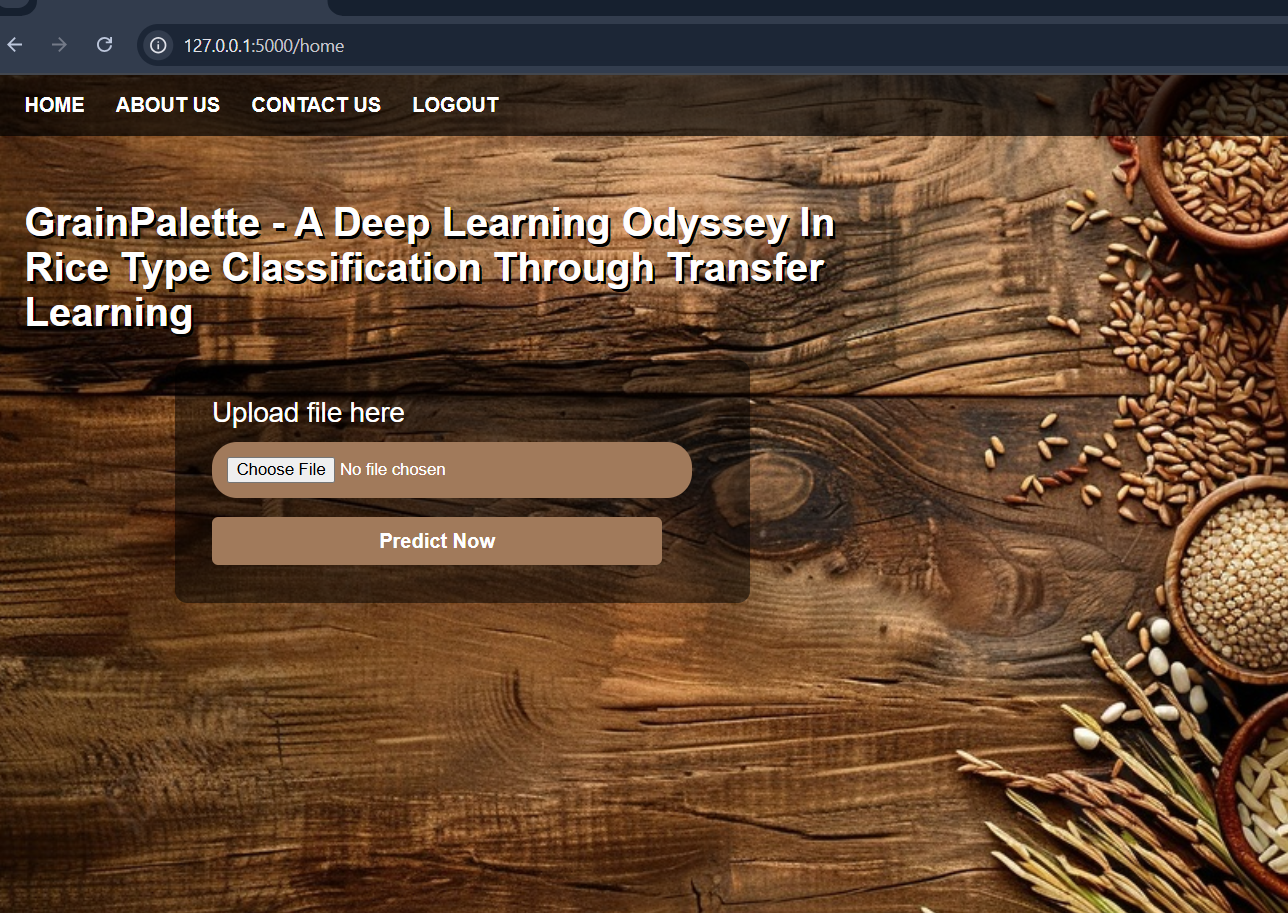
· **Login page** → authenticates the user.

· **Session or JWT token** → keeps them logged in.

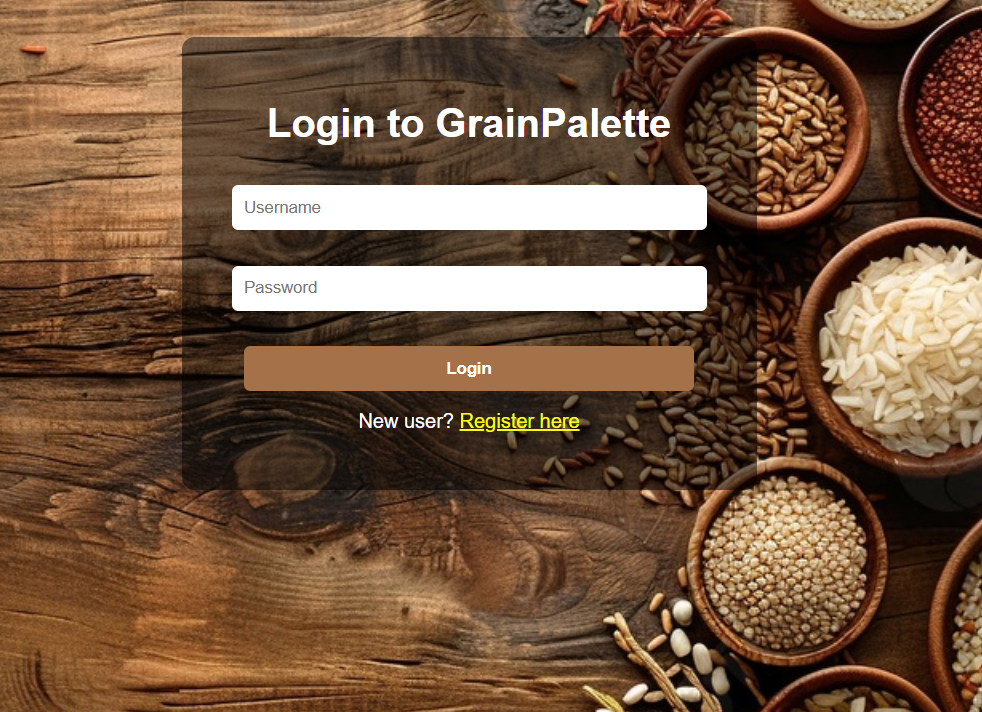
· **Restrict routes** (like /predict) → authorizes only logged-in users to access them.

· **Admin area (optional)** → special pages only accessible to admins.

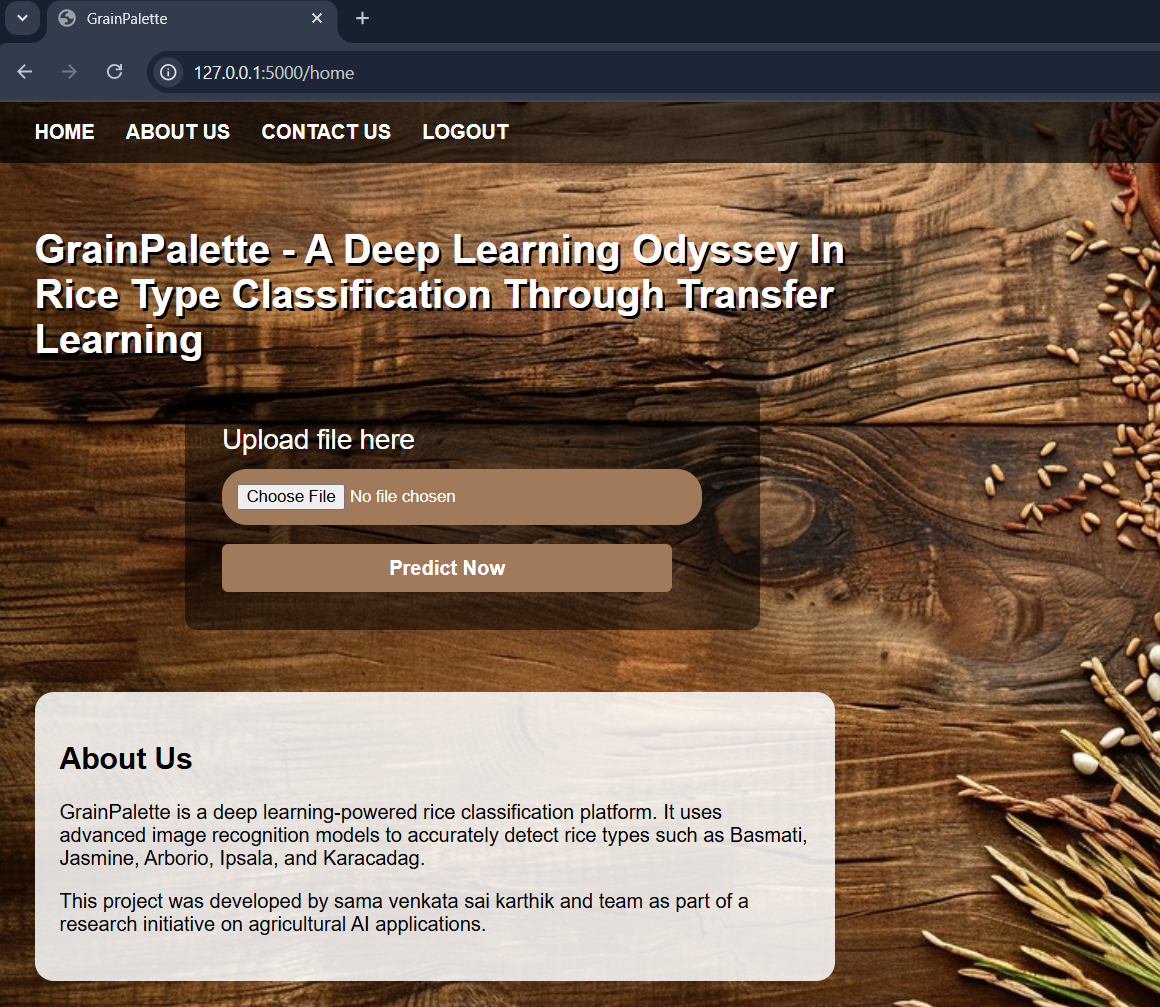
Home page(website):(home page&upload image )



User Login:



About Page:



**🧪 10. Testing**

**Tools Used:**

· Postman – For testing /predict endpoint

· Browser DevTools – Frontend inspection

· Pytest – Optional, for backend logic tests

**Testing Coverage:**

· Login/logout

· Prediction accuracy tests with known images

· Upload validations

· Page rendering tests

**📷 11. Demo or Showcase**

* **Screenshots**: As listed above
* **Optional Demo Video Link:** *[Insert YouTube/Google Drive demo link]*

**⚠️ 12. Known Issues**

· Limited dataset may affect rare rice types

· No user accounts or history in initial release

· No mobile app version yet

**🚀 13. Future Enhancements**

· Integrate rice disease detection module

· Add user account system with prediction history

· Support mobile app frontend

· Multi-language support for farmers

· Cloud deployment with API access

**📌 14. Conclusion**GrainPalette provides an innovative deep learning solution for rice variety classification, delivering high accuracy with a user-friendly web interface. It lays a solid foundation for automating rice quality control processes and supports scalable enhancements like disease detection or mobile platforms.