

YUGANTAR

PS1: AI-Powered Crop Disease Detection System

Problem Statement:

Crop diseases can reduce yields and cause major financial losses for farmers. Detecting diseases early and taking quick action can help prevent damage and improve crop health.

Build an AI-based system that analyzes crop images and environmental factors to identify diseases early. The system should give farmers real-time alerts, predictions, and useful advice on how to treat and prevent diseases.

Expected Solution:

A user-friendly mobile or web application that:

- Uses machine learning to detect crop diseases.
- Provides treatment suggestions based on identified diseases.
- Helps farmers protect their crops effectively.

Note: In the final evaluation, the accuracy and performance of the disease detection model will be given more weight than the graphical user interface (GUI). The primary focus should be on building a highly reliable and precise model before enhancing the UI.

Sample Dataset:

[Plant Diseases Dataset – Kaggle](#)

PS2: Timetable Scheduling System

Problem Statement:

Creating and managing a department timetable is a meticulous task that requires careful planning to avoid conflicts in faculty schedules, lecture room allocations, and lab sessions. The timetable coordinator plays a crucial role in ensuring a smooth academic schedule, but the process becomes challenging when frequent revisions are needed due to faculty changes, batch adjustments, or other unforeseen factors.

Develop an **AI-powered timetable scheduling system** for a **single department** (e.g., **Computer Science, Civil Engineering, Information Technology, AIML**). The system should streamline the scheduling process, optimize faculty and resource allocation, and allow seamless modifications when needed.

Expected Solution:

A timetable generation system that:

- Prevents conflicts in lecture rooms, labs, and faculty schedules.
- Ensures an even faculty workload across subjects.
- Allows quick revisions to accommodate schedule changes.
- Assigns 2 free lectures per batch per week while maintaining subject coverage.
- Notifies the coordinator if an optimized timetable cannot be generated.
- Balances lab and theory sessions for effective learning.

Output:

The system should generate three types of timetables:

1. **Lab Timetable** – Scheduling of practical sessions in labs.
2. **Individual Teacher Timetable** – Faculty-specific schedule showing assigned subjects and lecture times.
3. **Class Timetable** – A structured schedule for each batch, ensuring smooth academic operations.

Sample Timetables:[Sample Timetables](#)**Constraints to Follow:****For 2nd-Year Students:**

- College hours: **9 AM - 4 PM**
- Lunch break: **12 PM - 1 PM**
- Number of batches: **2**
- Subjects: **8**
- 2 free lectures per week per batch

For 3rd-Year Students:

- College hours: **10 AM - 5 PM**
- Lunch break: **1 PM - 2 PM**
- Number of batches: **2**
- Subjects: **5**
- 2 free lectures per week per batch

Faculty Constraints:

- Number of teachers: **7 to 10**
- Each teacher can teach a **minimum of 1 and a maximum of 3 subjects**

Resource Availability:

- Lecture rooms: **4**
- Labs: **4**
- For tutorial sessions, each batch should go to **different lecture rooms simultaneously**

Note: While these are the given constraints, a fully dynamic timetable generation system that can adapt to different inputs and changing requirements will be awarded higher marks in the final evaluation.

PS3: Local Employment Bridging Platform for MSME-Trained Candidates

Problem Statement:

Finding employment opportunities in local areas can be challenging for MSME-trained individuals due to a lack of centralized platforms catering to district, taluka, and village-level job seekers. A dedicated system can bridge the gap between skilled candidates and employers by offering job listings, career guidance, and skill-based employment opportunities.

Develop a **web or mobile application** that serves as a dedicated **job portal** for MSME-trained students and individuals, connecting them with relevant job openings across government, ITI, and local industries. The platform should be **intuitive, accessible, and multilingual**, ensuring that job seekers from all backgrounds can easily explore and apply for opportunities.

Expected Solution:

A feature-rich web or mobile application that includes:

- Job listings categorized by district with detailed descriptions (job title, salary, incentives, employer details, etc.).
- A multilingual interface to enhance accessibility.
- A **"Read Out Loud" feature**, enabling users who cannot read to listen to job descriptions and instructions.
- Skill-based job recommendations to help candidates find roles aligned with their training.
- Promotion of skilling courses by showcasing relevant job opportunities, motivating students to enhance their skills.