Predictive Maintenance System Prototype Documentation

Table of Contents

1. Introduction

This document outlines the prototype of a Predictive Maintenance System, which is designed to forecast the maintenance needs of machines based on various input parameters. The system combines a user-friendly frontend interface with a robust backend infrastructure to deliver accurate maintenance predictions, manage user authentication, and handle maintenance schedules.

2. System Overview/ repository structure

The Predictive Maintenance System aims to:

- Predict when a machine will need maintenance.

- Provide a user-friendly interface for data input and result display.

- Allow users to manage their maintenance schedules and history.

Features:

- User Signup and Login

- Predictive Maintenance Results

- Passbook Management

- Maintenance Schedule Calendar

Technology Stack:

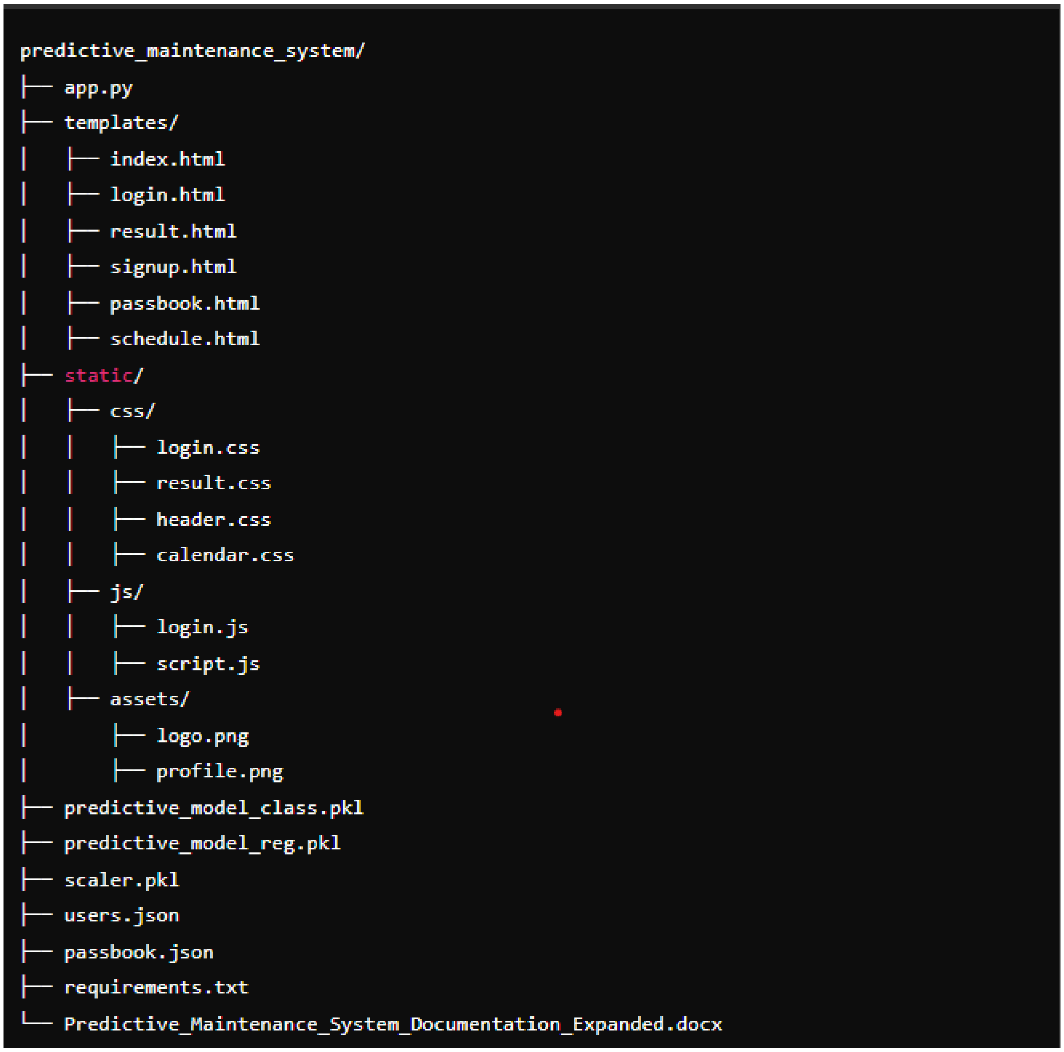
- Frontend: HTML, CSS, Bootstrap, JavaScript

- Backend: Flask, Python

- Database: JSON files for simplicity (users.json, passbook.json)

- Machine Learning: Scikit-learn, Joblib for model serialization

Repository structure



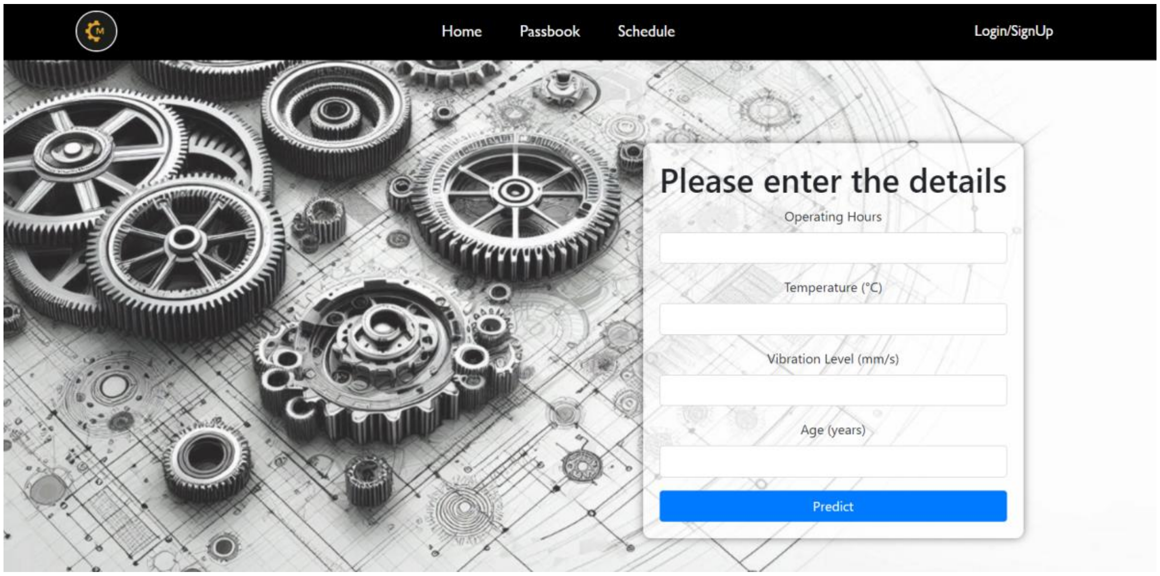
```
predictive_maintenance_system/
├── app.py
├── templates/
|   ├── index.html
|   ├── login.html
|   ├── result.html
|   ├── signup.html
|   ├── passbook.html
|   ├── schedule.html
├── static/
|   ├── css/
|   |   ├── login.css
|   |   ├── result.css
|   |   ├── header.css
|   |   ├── calendar.css
|   ├── js/
|   |   ├── login.js
|   |   ├── script.js
|   ├── assets/
|       ├── logo.png
|       ├── profile.png
├── predictive_model_class.pkl
├── predictive_model_reg.pkl
├── scaler.pkl
├── users.json
├── passbook.json
├── requirements.txt
└── Predictive_Maintenance_System_Documentation_Expanded.docx
```

3. Frontend Design

The frontend is designed using HTML, CSS, and Bootstrap to ensure a responsive and visually appealing interface. The key pages include:**Index Page**

Description: Initial landing page for the application.

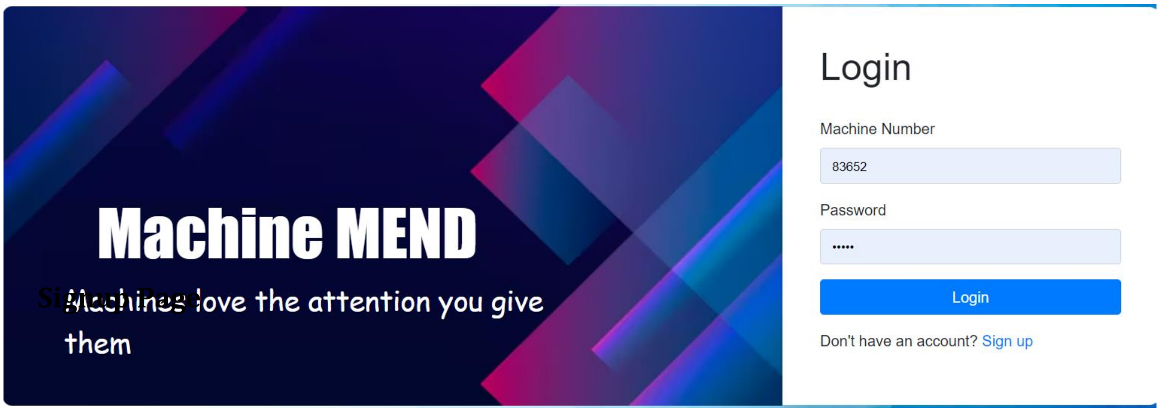
Elements: Navigation bar, information section, form for prediction input.



Login Page

Description: Interface for user login.

Elements: Form with fields for machine number and password, login button.


Machine MEND
Machines love the attention you give them
Login

Machine Number

83652

Password

•••••

Login
Don't have an account? Sign up

Signup Page

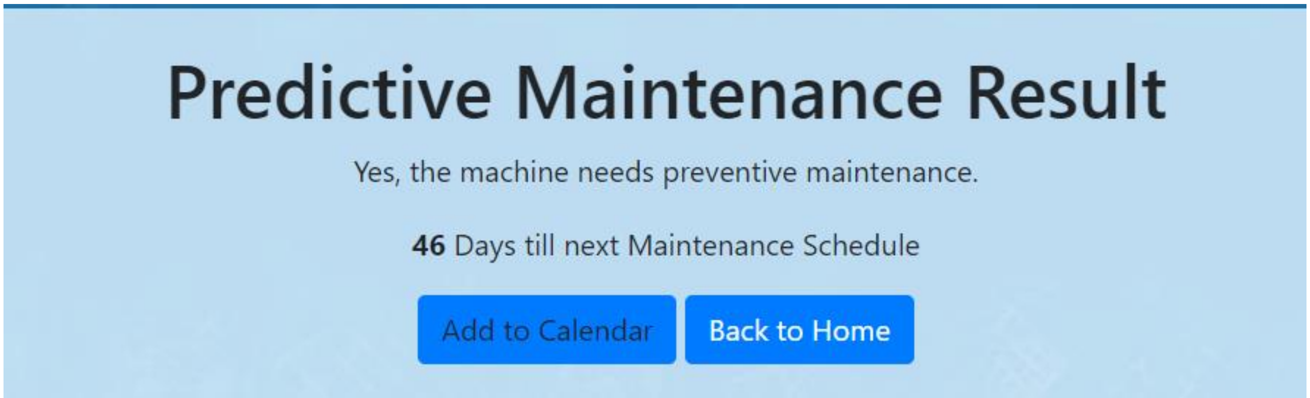
Description: Interface for user registration.

Elements: Form with fields for machine number and password, signup button.

Result Page

Description: Displays the prediction results.

Elements: Prediction message, expected days to maintenance, options to add to calendar and return to home.



Predictive Maintenance Result

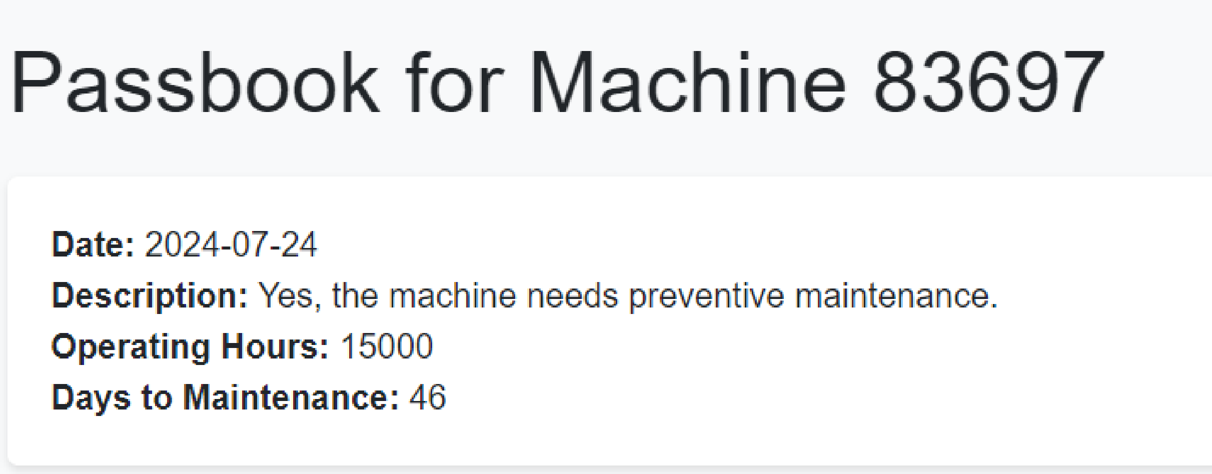
Yes, the machine needs preventive maintenance.

46 Days till next Maintenance Schedule

Passbook Page

Description: Displays maintenance history for the logged-in user.

Elements: List of maintenance events, dates, and details.



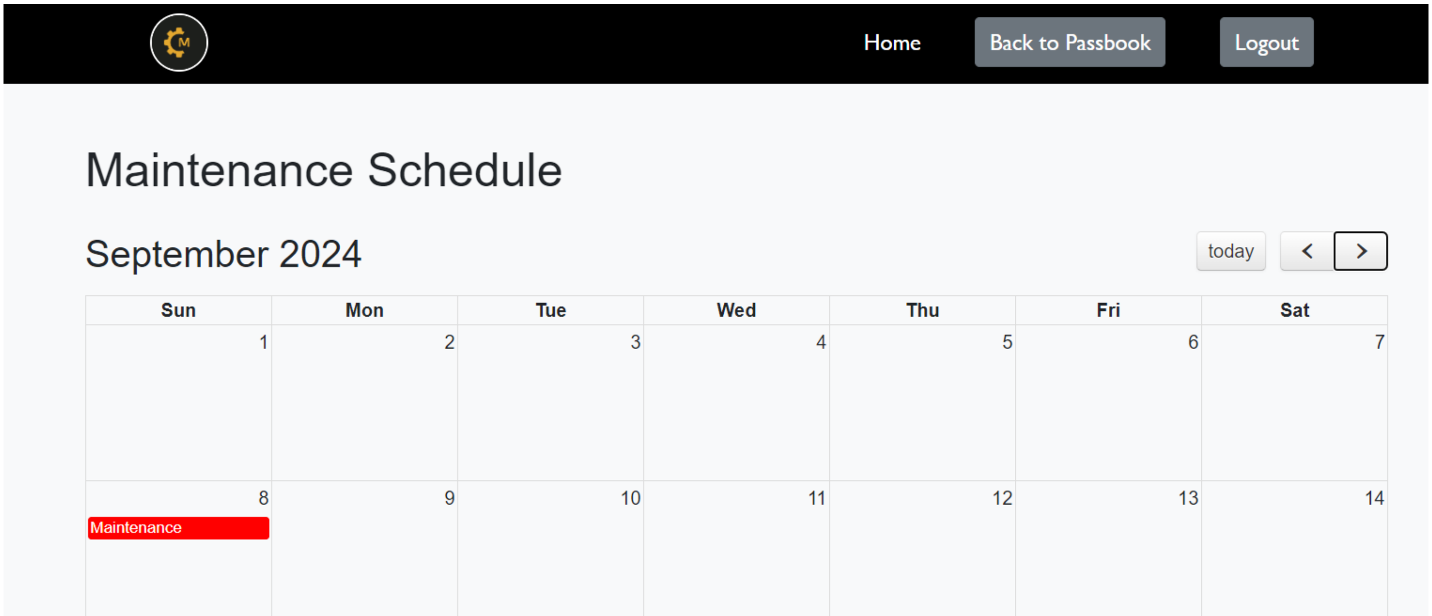
Passbook for Machine 83697

Date: 2024-07-24
Description: Yes, the machine needs preventive maintenance.
Operating Hours: 15000
Days to Maintenance: 46

Schedule Page

Description: Displays a calendar with maintenance dates marked.

Elements: Monthly calendar view, maintenance dates highlighted.



Maintenance Schedule

September 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8 Maintenance	9	10	11	12	13	14

4. Backend Design

The backend is developed using Flask, a lightweight web framework for Python. The backend handles:

User Authentication

- Signup, Login, and Logout functionality.

Prediction Processing

- Collects input data, preprocesses it, and uses machine learning models to generate predictions.

Passbook Management

- Stores and retrieves maintenance events for each user.

Schedule Calendar

- Displays a calendar with marked maintenance dates.

Dependencies:

- Flask: Web framework

- Pandas: Data manipulation

- Joblib: Model serialization

- JSON: Data storage

- Datetime: Date and time handling

Key Files:

- app.py: Main application file.

- predictive_model_class.pkl: Classification model for maintenance prediction.

- predictive_model_reg.pkl: Regression model for days to maintenance prediction.

- scaler.pkl: Scaler for input data preprocessing.

- users.json: Stores user data.

- passbook.json: Stores passbook entries.

5. User Authentication

User authentication is managed using machine number and password. The system supports:

Signup:

- Registers new users by storing their machine number and hashed password in users.json.

Login:

- Authenticates existing users by validating their credentials against the stored data.

Logout:

- Ends user sessions and clears session data.

Implementation:

- User data is stored in users.json.

- Sessions are managed using Flask’s session management with a generated secret key.

- Passwords are hashed using the werkzeug.security module.

6. Predictive Maintenance

The predictive maintenance feature uses trained machine learning models to predict:

Maintenance Need:

- Whether maintenance is needed (classification).

Days to Maintenance:

- Number of days until maintenance is required (regression).

Implementation:

- Input data is collected from the user via a form.

- Data is preprocessed using a scaler.

- Predictions are made using classification and regression models stored in predictive_model_class.pkl and predictive_model_reg.pkl.

7. Passbook

The passbook feature tracks maintenance events for each user, storing:

Event Date

- Date of the maintenance event.

Description

- Description of the maintenance performed.

Days to Maintenance

- Predicted number of days to the next maintenance.

Implementation:

- Passbook entries are stored in passbook.json.

- Users can view their passbook entries after logging in.

8. Schedule Calendar

The schedule calendar feature provides a visual representation of maintenance dates.

Implementation:

- Displays a monthly calendar view.

- Highlights maintenance dates based on the predictions.

- Users can view up to 20 future maintenance dates.

9. Installation Instructions

Prerequisites:

- Python 3.x

- Flask

- Pandas

- Joblib

Steps:

1. Clone the repository.

2. Install required packages:

```
pip install flask pandas joblib
```

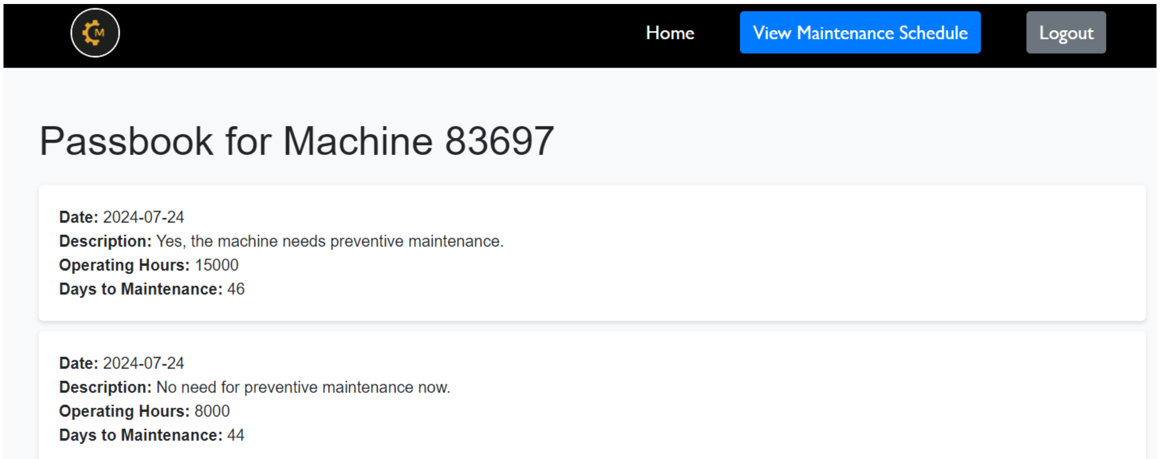
3. Run the application:

```
python app.py
```

10. Conclusion

This documentation outlines the architecture and functionality of the Predictive Maintenance System prototype. The system integrates user authentication, predictive analytics, and data management to provide a comprehensive maintenance tracking solution.

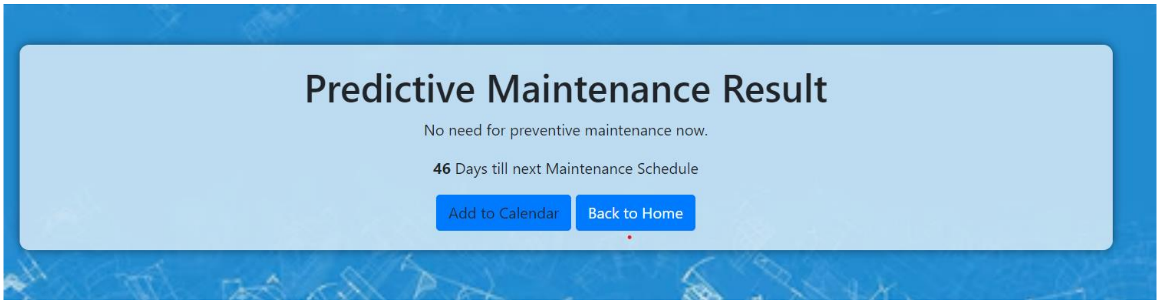
11. Output Samples



Passbook for Machine 83697

Date: 2024-07-24
Description: Yes, the machine needs preventive maintenance.
Operating Hours: 15000
Days to Maintenance: 46

Date: 2024-07-24
Description: No need for preventive maintenance now.
Operating Hours: 8000
Days to Maintenance: 44



Predictive Maintenance Result

No need for preventive maintenance now.

46 Days till next Maintenance Schedule