Laptop Price Prediction - Project Documentation

1. Project Overview

This project is a Full-Stack Web Application used to predict laptop prices based on various hardware specifications. It uses a Machine Learning model trained on historical data, served via a Flask backend, and presented through a modern, responsive frontend.

2. Backend Core (Python/Flask)

app.py
- The heart of the application.
- Initializes the Flask web server.
- Connects to the SQLite database (SQLAlchemy).
- Loads ML models (pickle) at startup.
- Defines API Routes:
  * /predict: Receives data, scales it, runs model, returns price.
  * /auth routes: Handles Login, Signup, OTP verification.
  * /history: Fetches user's past predictions.

otp_config.py
- Stores sensitive configuration for the OTP system.
- Contains SMTP credentials (for email sending) and expiry time settings.

3. Database & Storage

instance/laptop_price.db
- A binary SQLite database file.
- Stores structured data in tables: Users, Predictions, OTPs.
- Lightweight, serverless database ideal for this application.

view_database.py
- A custom utility script created for administration.
- Connects to the .db file and prints readable tables to the console.
- Useful for debugging and verifying data storage.

4. Machine Learning Assets

model.pkl, encoder.pkl, scaler.pkl
- model.pkl: The serialized Machine Learning regressor model.
- encoder.pkl: Converts categorical text (e.g., 'Dell', 'HP') into numbers.
- scaler.pkl: Standardizes numerical inputs (RAM, Weight) to match training scale.

5. Frontend (User Interface)

templates/landing.html
- The 'Welcome' page.
- Features animations and 'Get Started' call-to-action.
- Designed to impress users immediately.

templates/index.html
- The main dashboard.

- Contains the prediction form (Brand, RAM, CPU, etc.).
- Displays the Price Prediction History table.

static/js/app.js
- Handles client-side logic.
- Validates form inputs before sending.
- Makes asynchronous (AJAX) calls to the Flask backend.
- Dynamically updates the UI with the predicted price without reloading.