# ESTIMATING STOCK KEEPING UNIT USING ML

**Final Model Deployment: Flask Web Application**

**Objective:**

To build and deploy a user-friendly web application that allows users to input recent sales data and receive predicted demand values for a stock keeping unit (SKU).

**Deployment Stack:**

- Backend: Flask (Python micro web framework)
- Frontend: HTML/CSS
- Model Serialization: `pickle`
- Environment: VS CODE

---

**Backend (app.py):**

The backend is responsible for loading the trained model, processing user inputs, generating features, and producing predictions.

Key Components:

`app.py`

- defines the Flask application
- Loads `sales_demand_forecasting.pkl` using `pickle`
- Receives form data (sales from last 4 days)
- Calculates derived features (rolling mean, expanding mean, interaction terms)
- Supplies default values for encoded fields and price info
- Sends input to the model for prediction
- Returns output to a results page

Flask Routes:

- `/` — Loads the index form page
- `/y_predict` — Handles POST request and returns the result

**Input Fields Handled in Backend:**

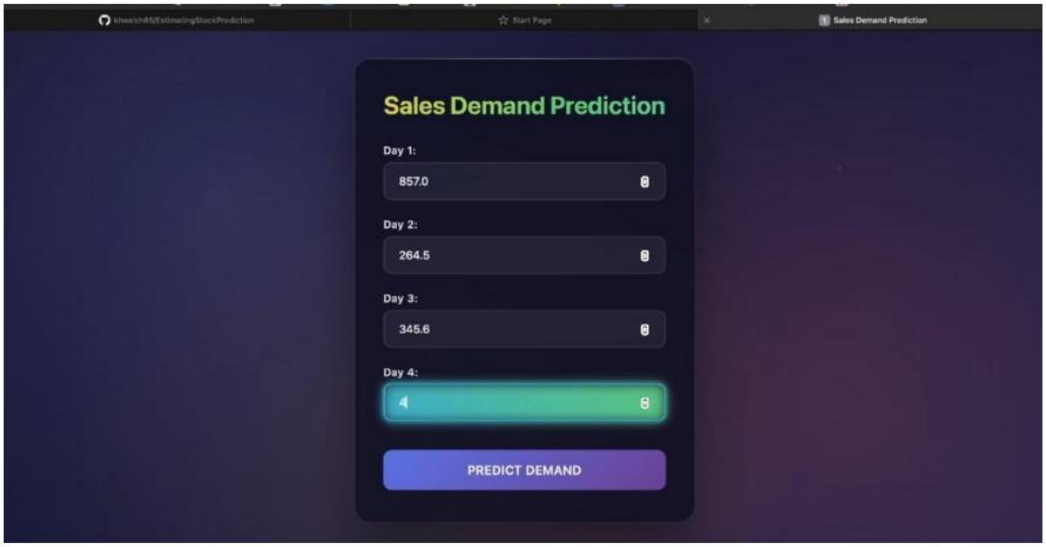
- `day_1`, `day_2`, `day_3`, `day_4`

---

**Frontend (HTML Pages):**

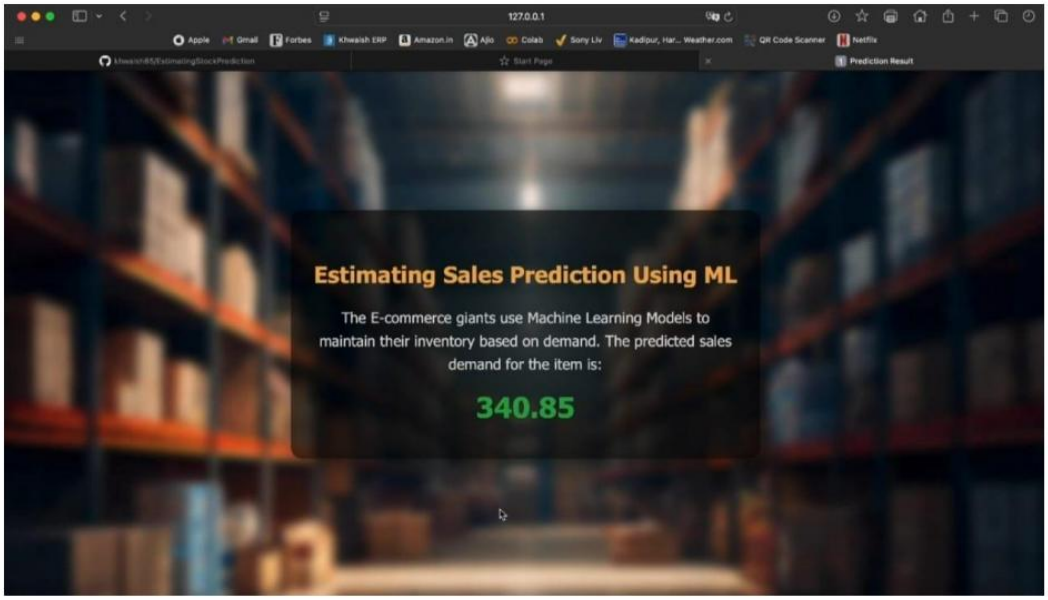
**1. index.html**

- Collects user inputs for 4 recent days of SKU sales
- Includes glowing input effects for modern UI design
- Submit button posts form data to `/y_predict`




**2. result.html**

- Displays the predicted units sold returned by the model
- Styled with a warehouse-themed background and overlay card
- Includes interpretation text on ML usage in retail



**Conclusion:**

This deployment setup bridges data science and application usability, enabling real-time sales forecasting with minimal user effort.