E-Commerce Customer Churn Prediction

ChurnAI is a full-stack machine learning application designed to predict customer churn in an e-commerce platform. The system uses a trained Random Forest model to classify customers based on their behaviour and provides a clean, interactive frontend for business users.

This repository contains:

- **Backend**: Flask REST API for authentication and churn prediction
- **Frontend**: React + Vite web interface
- **ML Model**: Pre-trained Random Forest churn prediction model

➢ Features

- User authentication (Admin login & Demo login)
- Customer churn prediction using Machine Learning
- Real-time predictions via REST API
- Pre-trained Random Forest model (rf_churn_model.pkl)
- Modern React frontend with Vite
- CORS-enabled backend for local development

➢ Tech Stack

Backend

- Python 3.x
- Flask
- Flask-CORS
- Pandas, NumPy
- Joblib (for model loading)

Frontend

- React
- Vite
- JavaScript / HTML / CSS

- **Project Structure**

churnai-main/

```
│
├── backend/
│   ├── app.py              # Flask API
│   ├── requirements.txt    # Backend dependencies
│   └── rf_churn_model.pkl  # Trained ML model
│
├── frontend/
│   ├── src/                # React source code
│   ├── public/             # Static assets
│   ├── index.html
│   ├── package.json
│   └── package-lock.json
│
├── README.md               # Project documentation
└── SPLINE_SETUP.md         # Optional spline animation setup
```

- **Setup Instructions**

1. Clone / Extract the Project

git clone <repository-url>

cd churnai-main

Or extract the ZIP file and navigate to the project folder.

- **Backend Setup (Flask API)**

Step 1: Navigate to Backend Folder

cd backend

Step 2: Create Virtual Environment (Optional but Recommended)

python -m venv venv

venv\Scripts\activate # Windows

source venv/bin/activate # Linux/Mac

Step 3: Install Dependencies

pip install -r requirements.txt

Step 4: Run the Backend Server

python app.py

The backend will start at:

http://127.0.0.1:5000

- **Frontend Setup (React + Vite)**

Step 1: Navigate to Frontend Folder

cd frontend

Step 2: Install Node Modules

npm install

Step 3: Start Development Server

npm run dev

The frontend will run at:

http://localhost:5173

- **Authentication Options**

1. Demo Login

- Click **Demo Login** on the UI
- No credentials required

2. Admin Login

Username: admin

Password: admin

- **API Endpoints**

Login

POST /api/auth/login

Options:

- Demo login
- Username & password authentication

- **Churn Prediction**

POST /api/predict

Requires authentication

Input Parameters (Sample):

- Tenure
- City Tier
- Hour Spend On App
- Satisfaction Score
- Order Count
- Cashback Amount
- Preferred Payment Mode
- Gender, Marital Status, Order Category

Output:

- Churn probability
- Churn / Non-Churn classification

- **Machine Learning Model**

- Algorithm: Random Forest Classifier
- Model file: rf_churn_model.pkl
- Handles categorical and numerical features
- Optimized for customer behaviour patterns

- **Notes & Tips**
  - Ensure backend is running before starting frontend
  - Use Chrome or Edge for best UI experience
  - CORS is pre-configured for local development

- **Future Enhancements**
  - Add SHAP-based explainability
  - Role-based user access
  - Cloud deployment (AWS / Azure)
  - Database integration
  - Real-time analytics dashboard

- **Author**

Developed as part of an E-Commerce Customer Churn Prediction project using Machine Learning and Full-Stack development.

- **License**