## System Architecture

### Components

1. **Data Ingestion**:
   * Source: Kaggle dataset
   * Preprocessing: Cleaning, encoding, scaling
2. **Model Training**:
   * Algorithm: XGBoost
   * Training and tuning
3. **Web Application**:
   * Framework: Flask (or Django)
   * Frontend: HTML, CSS, JavaScript
4. **Prediction Service**:
   * Endpoint: /predict
   * Input: 23 feature values
   * Output: Prediction result

### Data Flow

1. **User Input**:
   * User inputs data through the web interface.
   * Data is sent to the backend server.
2. **Backend Processing**:
   * The server processes the data and feeds it into the trained model.
   * The model returns the prediction result.
3. **Output Display**:
   * The result is displayed to the user.

### **Description**

1. **User Interface**:
   * The user interface collects user inputs and displays predictions.
   * Built using HTML, CSS, and JavaScript.
2. **Backend Server**:
   * The server processes input data, interacts with the model, and returns predictions.
   * Built using Flask (Python).
3. **Model**:
   * The XGBoost model is trained on preprocessed data to predict defaults.
   * The model is loaded by the Flask application for making predictions.

### Diagram

+------------------+ +----------------------+ +-----------------+

| | | | | |

| User Interface +------->| Backend Server +------->| Model |

| | | (Flask Application) | | (XGBoost) |

| | | | | |

+------------------+ +----------------------+ +-----------------+