Project Proposal: Real-Time 1xBet Prediction App

Objective

Develop a web application that predicts match outcomes and provides real-time betting suggestions for 1xBet using machine learning. The app will deliver insights and recommendations to users based on historical data, live match stats, and ML predictions.

Tech Stack

- **Frontend:** React.js, Tailwind CSS, Axios (for API calls), Socket.IO (for real-time updates)
- **Backend:** Python (Flask/FastAPI), Pandas, NumPy, Scikit-learn, TensorFlow/PyTorch (ML model), Socket.IO
- **Database:** MongoDB/PostgreSQL for storing historical match data and predictions
- **Hosting/Deployment:** Heroku / AWS / Vercel (frontend), Python backend on AWS/Heroku

Project Structure

```
1xBetPredictionApp/
├─ backend/
│  ├─ app.py             # Flask/FastAPI main server
│  ├─ model/
│  │   ├─ trainer.py     # Train ML models
│  │   └─ predictor.py   # Load model & predict outcomes
│  ├─ data/
│  │   └─ historical_matches.csv
│  ├─ utils/
│  │   └─ data_processing.py
│  └─ requirements.txt
│
├─ frontend/
│  ├─ src/
│  │   ├─ components/
│  │   ├─ pages/
│  │   ├─ services/
│  │   │   └─ api.js      # Axios calls to backend
│  │   └─ App.jsx
│  ├─ package.json
│  └─ tailwind.config.js
│
└─ README.md
```

Key Features

1. **Real-Time Predictions**
   - Fetch live match data via APIs
   - Predict likely outcomes using trained ML models

2. **Historical Analysis**
    - Use past match data to improve prediction accuracy
3. **Betting Suggestions**
    - Suggest odds with highest probability of success
    - Highlight low-risk bets
4. **User Interface**
    - Clean React UI
    - Display predictions, statistics, and recommendations
    - Real-time notifications for live matches
5. **ML Model**
    - Train on historical match data (teams, players, odds, scores)
    - Use classification/regression algorithms (Logistic Regression, XGBoost, Neural Networks)

Workflow

1. Backend fetches **live match data**
2. ML model predicts match outcome probabilities
3. Backend sends **real-time suggestions** via API/Socket.IO
4. React frontend displays results to users

Outcome

- Users can view **real-time betting suggestions**
- Provides **data-driven insights** for smarter betting decisions
- Modular structure allows **easy model upgrades**