Data Analytics Portfolio Project Setup Guide

Table of Contents

Infrastructure Setup

Current Status

- Frontend: Vite React app deployed on Netlify
- GitHub: Repository created and connected
- Next steps: Setting up backend and database

Backend Setup on Render.com

1. Create a new account on Render.com
2. Create a new Web Service
    - Connect to your GitHub repository
    - Select Python environment
    - Set build command: `pip install -r requirements.txt`
    - Set start command: `uvicorn main:app --host 0.0.0.0 --port $PORT`
    - Add environment variables:

```
DATABASE_URL=your-supabase-url
CORS_ORIGIN=your-frontend-url
```

3. Required files in repository:

```
backend/
├── requirements.txt
├── main.py
└── .gitignore
```

4. Content for requirements.txt:

```
fastapi
uvicorn
pandas
scikit-learn
python-dotenv
sqlalchemy
psycopg2-binary
```

Database Setup on Supabase

1. Create new project on Supabase
2. Set up tables:

```
-- games table
CREATE TABLE games (
    id SERIAL PRIMARY KEY,
    game_date DATE,
    attendance INTEGER,
    ticket_price DECIMAL(10,2),
    day_of_week VARCHAR(10),
    temperature DECIMAL(5,2),
    precipitation DECIMAL(5,2),
    opponent VARCHAR(100),
    promotion VARCHAR(100),
    season VARCHAR(4)
);
```

3. Save connection string for backend configuration

Frontend Deployment

You can keep your frontend on Netlify - no need to switch to Vercel. Netlify provides all necessary features for this project:

- Automatic deploys from Git
- Environment variable management
- Custom domains
- HTTPS encryption

Connecting Services

1. Configure CORS and Environment Variables

Backend (main.py):

```
from fastapi.middleware.cors import CORSMiddleware

app.add_middleware(
    CORSMiddleware,
    allow_origins=[os.getenv("CORS_ORIGIN")],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

Frontend (.env):

```
VITE_API_URL=https://your-render-app.onrender.com
```

2. Testing Connection

1. Create a test endpoint in backend:

   ```
   @app.get("/api/health")
   async def health_check():
       return {"status": "healthy"}
   ```

2. Test in frontend:

   ```
   const testConnection = async () => {
     try {
       const response = await fetch(
         `${import.meta.env.VITE_API_URL}/api/health`
       );
       const data = await response.json();
       console.log('Connection test:', data);
     } catch (error) {
       console.error('Connection error:', error);
     }
   };
   ```

Skills Development Checklist

Technical Skills

1. **Data Processing**
   - ☐ CSV file handling
   - ☐ Data cleaning
   - ☐ Missing value handling
   - ☐ Data type conversion
   - ☐ Date/time manipulation
2. **SQL Skills**
   - ☐ Basic CRUD operations
   - ☐ Aggregations and grouping
   - ☐ Window functions
   - ☐ Joins across tables
   - ☐ Complex filtering
3. **Python Skills**

- [ ] Pandas operations
- [ ] Statistical analysis
- [ ] Data transformation
- [ ] Machine learning basics
- [ ] API development

4. **Visualization**
   - [ ] Interactive charts
   - [ ] Multiple chart types
   - [ ] Dashboard layout
   - [ ] User filters
   - [ ] Responsive design

Analysis Skills

1. **Business Understanding**
   - [ ] KPI identification
   - [ ] Trend analysis
   - [ ] Seasonality detection
   - [ ] Revenue impact analysis
2. **Statistical Analysis**
   - [ ] Correlation analysis
   - [ ] Hypothesis testing
   - [ ] Regression analysis
   - [ ] Predictive modeling
3. **Communication**
   - [ ] Data storytelling
   - [ ] Visual presentation
   - [ ] Technical documentation
   - [ ] Recommendations

Example Analysis Questions

1. "What factors most strongly influence game attendance?"
2. "Can we predict attendance for upcoming games?"
3. "How do promotional events impact revenue?"
4. "What is the optimal pricing strategy by season?"
5. "Is there a correlation between weather and attendance?"

Original Conversation Summary

The project began with creating a sports analytics dashboard for a fictional minor league baseball team called "Metro Miners". The initial implementation included:

1. A React component using Recharts for visualization
2. Sample data structure with game attendance, pricing, and weather
3. Basic filtering by season
4. Responsive layout with Tailwind CSS

The goal is to demonstrate key data analytics skills required for a position similar to the provided job description from Demand Analytics, focusing on:

- Data processing and analysis
- Statistical modeling
- Visualization
- Business intelligence
- Technical implementation

To continue this project in a new Claude conversation, use this prompt: "I am building a sports analytics portfolio project. I have my frontend deployed on Netlify, backend planned for Render.com, and database on Supabase. The project demonstrates data analytics skills through analyzing baseball game attendance, pricing, and weather data. Please help me [specific next step you want to work on]."