Folder Structure

```
ai-race-engineer/
│
├── data/                          # Data storage and management
│   ├── raw/                       # Raw FastF1 cache and downloads
│   ├── processed/                 # Cleaned and feature-engineered data
│   ├── datasets/                  # Training datasets
│   └── scripts/
│       ├── fetch_fastf1_data.py       # Download race sessions
│       ├── preprocess_telemetry.py    # Clean and align telemetry
│       └── generate_features.py       # Feature engineering for ML
│
├── models/                        # ML model training and inference
│   ├── tire_degradation/
│   │   ├── train.py               # LSTM/GRU training
│   │   ├── model.py               # Model architecture
│   │   ├── inference.py           # Prediction pipeline
│   │   └── checkpoints/           # Saved models
│   ├── pit_strategy/
│   │   ├── train.py               # Classification model
│   │   ├── model.py
│   │   └── checkpoints/
│   └── verstappen_simulator/
│       ├── driving_style_model.py     # Style-specific adjustments
│       └── comparison.py              # Aggressive vs conservative
│
├── agents/                        # Multi-agent system
│   ├── base_agent.py              # Abstract agent class
│   ├── telemetry_agent.py         # Analyzes telemetry data
│   ├── strategy_agent.py          # Pit strategy decisions
│   ├── communication_agent.py         # Natural language generation
│   ├── verstappen_agent.py            # Verstappen-specific reasoning
│   ├── meta_controller.py             # Agent orchestration
```

```
│   └── prompts/                    # LLM prompts for each agent
│       ├── telemetry_prompts.py
│       ├── strategy_prompts.py
│       └── communication_prompts.py
│
├── backend/                        # API and services
│   ├── app.py                      # FastAPI application
│   ├── routes/
│   │   ├── telemetry.py            # Telemetry endpoints
│   │   ├── strategy.py             # Strategy endpoints
│   │   └── simulation.py           # Run simulations
│   ├── services/
│   │   ├── agent_service.py        # Agent interaction logic
│   │   ├── prediction_service.py   # ML inference
│   │   └── cache_service.py        # Redis caching
│   ├── schemas/                    # Pydantic models
│   │   ├── telemetry_schema.py
│   │   └── strategy_schema.py
│   └── workers/
│       └── celery_worker.py        # Background tasks
│
├── frontend/                       # Web interface
│   ├── src/
│   │   ├── components/
│   │   │   ├── TelemetryChart.tsx  # Telemetry visualization
│   │   │   ├── StrategyPanel.tsx   # Strategy recommendations
│   │   │   ├── ChatInterface.tsx   # Talk to race engineer
│   │   │   └── ComparisonView.tsx  # Verstappen vs baseline
│   │   ├── pages/
│   │   │   ├── index.tsx           # Home page
│   │   │   ├── session.tsx         # Session analysis
│   │   │   └── simulator.tsx       # Live simulation
│   │   ├── services/
│   │   │   └── api.ts              # API client
│   │   └── utils/
│   │       └── telemetry_utils.ts  # Data transformation
```

```
│   ├── public/
│   └── package.json
│
├── notebooks/                    # Jupyter notebooks
│   ├── 01_data_exploration.ipynb      # EDA on FastF1 data
│   ├── 02_tire_degradation_analysis.ipynb
│   ├── 03_verstappen_style_analysis.ipynb
│   └── 04_agent_testing.ipynb         # Test agent interactions
│
├── tests/                        # Testing
│   ├── test_agents/
│   ├── test_models/
│   ├── test_api/
│   └── test_integration/
│
├── config/                       # Configuration files
│   ├── agent_config.yaml             # Agent parameters
│   ├── model_config.yaml             # Model hyperparameters
│   └── api_config.yaml               # API settings
│
├── docker/                       # Docker configuration
│   ├── Dockerfile.backend
│   ├── Dockerfile.frontend
│   └── docker-compose.yml
│
├── mlops/                        # MLOps utilities
│   ├── mlflow_tracking.py            # Experiment tracking
│   ├── model_registry.py             # Model versioning
│   └── monitoring.py                 # Performance monitoring
│
├── scripts/                      # Utility scripts
│   ├── setup_environment.sh          # Environment setup
│   └── download_all_data.py          # Batch data download
│
├── requirements.txt                  # Python dependencies
├── pyproject.toml                    # Poetry configuration (alternative)
```

```
├── .env.example                # Environment variables template
├── README.md                   # Project documentation
└── LICENSE
```