TAJIR PROJECT

Complete Project Documentation

Generated on: January 17, 2026 at 21:57

Table of Contents

1. Project Overview

TAJIR is a comprehensive trading and financial analysis platform with both backend and frontend components. It provides real-time data processing, AI-driven analysis, and WebSocket-based live updates for trading decisions.

Key Features:

■ Real-time WebSocket data streaming

■ AI-powered market analysis

■ Forex data integration

■ Technical analysis with pandas-ta

■ Live notification system

■ Cross-platform Flutter frontend

■■ Secure user authentication

2. Project Structure

- Backend/
  - READ.md
  - app/
    - __init__.py
    - api/
      - __init__.py
      - routes.py
      - websocket.py
    - enhanced_websocket_manager.py
    - example_usage.py
    - forex_data_service.py
    - live_updates_routes.py
    - main.py
    - models/
      - __init__.py
      - live_update.py
      - ml_task.py
    - services/
      - __init__.py
      - ai_analysis_service.py
      - connection_manager.py
      - forex_data_service.py
      - ml_processor.py
      - notification_service.py
      - trading_bot_service.py
    - users.py
    - utils/
      - __init__.py
    - websocket_manager.py
    - websocket_routes.py
  - main.py
  - pandas-ta-index.html
  - requirements.txt
  - run.py
  - test_live_updates.py
  - test_pandas_ta_import.py
  - test_ta.py
  - tests/
    - __init__.py
    - test_api.py
  - venv/
    - Include/
    - Lib/
      - site-packages/
    - Scripts/
      - Activate.ps1
      - activate
      - activate.bat
      - deactivate.bat
      - email_validator.exe
      - fastapi.exe
      - pip.exe
      - pip3.10.exe
      - pip3.exe
      - python.exe
      - pythonw.exe
      - uvicorn.exe

```
      ■ websockets.exe
    ■ pyvenv.cfg
■ Frontend/
  ■ README.md
  ■ analysis_options.yaml
  ■ android/
    ■ app/
      ■ build.gradle.kts
      ■ src/
    ■ build.gradle.kts
    ■ gradle/
      ■ wrapper/
    ■ gradle.properties
    ■ gradlew
    ■ gradlew.bat
    ■ local.properties
    ■ settings.gradle.kts
    ■ tajir_frontend_android.iml
  ■ assets/
    ■ images/
      ■ logo.png
  ■ ios/
    ■ Flutter/
      ■ AppFrameworkInfo.plist
      ■ Debug.xcconfig
      ■ Generated.xcconfig
      ■ Release.xcconfig
      ■ ephemeral/
      ■ flutter_export_environment.sh
    ■ Runner/
      ■ AppDelegate.swift
      ■ Assets.xcassets/
      ■ Base.lproj/
      ■ GeneratedPluginRegistrant.h
      ■ GeneratedPluginRegistrant.m
      ■ Info.plist
      ■ Runner-Bridging-Header.h
    ■ Runner.xcodeproj/
      ■ project.pbxproj
      ■ project.xcworkspace/
      ■ xcshareddata/
    ■ Runner.xcworkspace/
      ■ contents.xcworkspacedata
      ■ xcshareddata/
    ■ RunnerTests/
      ■ RunnerTests.swift
  ■ lib/
    ■ core/
      ■ config/
      ■ models/
      ■ theme/
      ■ utils/
      ■ widgets/
    ■ features/
      ■ ai_chat/
      ■ dashboard/
      ■ settings/
      ■ task_creation/
      ■ task_history/
    ■ helpers/
      ■ mock_data_helper.dart
    ■ main.dart
    ■ providers/
      ■ task_provider.dart
      ■ user_provider.dart
    ■ routes/
      ■ app_routes.dart
    ■ services/
      ■ api_service.dart
      ■ firebase_service.dart
      ■ gemini_service.dart
      ■ live_update_service.dart
      ■ websocket_service.dart
  ■ linux/
    ■ CMakeLists.txt
    ■ flutter/
      ■ CMakeLists.txt
      ■ ephemeral/
      ■ generated_plugin_registrant.cc
      ■ generated_plugin_registrant.h
      ■ generated_plugins.cmake
    ■ runner/
      ■ CMakeLists.txt
      ■ main.cc
      ■ my_application.cc
      ■ my_application.h
  ■ macos/
    ■ Flutter/
      ■ Flutter-Debug.xcconfig
      ■ Flutter-Release.xcconfig
      ■ GeneratedPluginRegistrant.swift
      ■ ephemeral/
    ■ Runner/
      ■ AppDelegate.swift
      ■ Assets.xcassets/
      ■ Base.lproj/
      ■ Configs/
      ■ DebugProfile.entitlements
      ■ Info.plist
      ■ MainFlutterWindow.swift
      ■ Release.entitlements
    ■ Runner.xcodeproj/
      ■ project.pbxproj
      ■ project.xcworkspace/
      ■ xcshareddata/
    ■ Runner.xcworkspace/
      ■ contents.xcworkspacedata
      ■ xcshareddata/
    ■ RunnerTests/
      ■ RunnerTests.swift
  ■ pubspec.lock
  ■ pubspec.yaml
  ■ tajir_frontend.iml
  ■ test/
    ■ widget_test.dart
  ■ venv/
    ■ Include/
    ■ Lib/
      ■ site-packages/
    ■ Scripts/
```

      - Activate.ps1
      - activate
      - activate.bat
      - deactivate.bat
      - pip.exe
      - pip3.10.exe
      - pip3.exe
      - python.exe
      - pythonw.exe
    - pyvenv.cfg
  - web/
    - favicon.png
    - icons/
      - Icon-192.png
      - Icon-512.png
      - Icon-maskable-192.png
      - Icon-maskable-512.png
    - index.html
    - manifest.json
  - windows/
    - CMakeLists.txt
    - flutter/
      - CMakeLists.txt
      - ephemeral/
      - generated_plugin_registrant.cc
      - generated_plugin_registrant.h
      - generated_plugins.cmake
    - runner/
      - CMakeLists.txt
      - Runner.rc
      - flutter_window.cpp
      - flutter_window.h
      - main.cpp
      - resource.h
      - resources/
      - runner.exe.manifest
      - utils.cpp
      - utils.h
      - win32_window.cpp
      - win32_window.h
- firebase.json
- generate_project_pdf.py
- package-lock.json
- package.json
- project-structure.txt
- public/
  - index.html
- requirements.txt
- y/
  - index.html
  - main.py
  - requirements.txt

3. Backend Architecture

The backend is built with Flask and provides RESTful APIs, WebSocket connections, and real-time data processing capabilities.

Backend Modules:

api: routes.py, websocket.py

models: live_update.py, ml_task.py

services: ai_analysis_service.py, connection_manager.py, forex_data_service.py, ml_processor.py, notification_service.py ... and 1 more

utils:

4. Frontend Architecture

The frontend is a cross-platform Flutter application providing intuitive UI for market analysis and trading operations.

Supported Platforms: iOS, Android, Web, Windows, macOS, Linux

5. Configuration Files

README files

File: Backend/READ.md

~~~
# ML Live Update Backend

Real-time machine learning backend with WebSocket-based live progress updates.

## Quick Start

1. **Setup virtual environment:**
```bash
   python -m venv .venv
   .venv\Scripts\activate  # Windows
   # source .venv/bin/activate  # Linux/Mac
```

2. **Install dependencies:**
```bash
   pip install -r requirements.txt
```

3. **Run server:**
```bash
   python run.py
```

4. **Access API docs:**
   - Swagger UI: http://localhost:8000/docs
   - ReDoc: http://localhost:800...
~~~

File: Frontend/README.md

````
# tajir_frontend

A new Flutter project.

## Getting Started

This project is a starting point for a Flutter application.

A few resources to get you started if this is your first Flutter project:

- [Lab: Write your first Flutter app](https://docs.flutter.dev/get-started/codelab)
- [Cookbook: Useful Flutter samples](https://docs.flutter.dev/cookbook)

For help getting started with Flutter development, view the
[online documentation](https://docs.flutter.dev/), which offers tutorials,
samples, g...
````

Configuration

File: package.json

```
{
  "dependencies": {
    "firebase": "^12.7.0"
  }
}
```

File: firebase.json

```
{
  "hosting": {
    "public": "y",
    "ignore": [
      "firebase.json",
      "**/.*",
      "**/node_modules/**"
    ],
    "rewrites": [
      {
        "source": "**",
        "destination": "/index.html"
      }
    ]
  }
}
```

File: Backend/requirements.txt

```
#■ ■C■o■r■e■ ■B■a■c■k■e■n■d■ ■R■e■q■u■i■r■e■m■e■n■t■s■ ■(■M■i■n■i■m■a■l■)■
■
■f■a■s■t■a■p■i■=■=■0■.■1■0■9■.■0■
■
■u■v■i■c■o■r■n■[■s■t■a■n■d■a■r■d■]■=■=■0■.■2■7■.■0■
■
■w■e■b■s■o■c■k■e■t■s■=■=■1■2■.■0■
■
■p■y■t■h■o■n■-■m■u■l■t■i■p■a■r■t■=■=■0■.■0■.■6■
■
■
■
■#■ ■D■a■t■a■b■a■s■e■
■
■m■o■t■o■r■=■=■3■.■3■.■2■
■
■p■y■m■o■n■g■o■=■=■4■.■6■.■1■
■
■
■
■#■ ■D■a■t■a■ ■P■r■o■c■e■s■s■i■n■g■
■
■p■a■n■d■a■s■=■=■2■.■1■.■4■
■
■n■u■m■p■y■=■=■1■.■2■6■.■3■
■
■p■a■n■d■a■s■-■t■a■=■=■0■.■4■.■7■1■b■0■
■
■
■
■#■ ■H■T■T■...
```

File: Frontend/pubspec.yaml

```
name: forex_companion
description: AI-powered Forex trading assistant
publish_to: 'none'
version: 1.0.0+1

environment:
  sdk: '>=3.0.0 <4.0.0'
dependencies:
  flutter:
    sdk: flutter

  # UI
  cupertino_icons: ^1.0.8

  # State Management
  provider: ^6.1.1

  # HTTP & WebSocket
  http: ^1.2.0
  web_socket_channel: ^3.0.3

  # Date Formatting
  intl: ^0.20.2

  # Firebase
  firebase_core: ^4.3.0
  firebase_auth: ^6.1.3
  cloud_firestore: ^6.1.1
  firebase_storage: ^13.0.5

  # Gemi...
```

.env-like

File: Backend/.env

```
# Server Configuration
HOST=0.0.0.0
PORT=8080
DEBUG=True

# Database (MongoDB)
MONGODB_URL=mongodb://localhost:27017
MONGODB_DB_NAME=forex_companion

# Firebase (if needed)
FIREBASE_PROJECT_ID=forexcompanion-e5a28

# API Keys
GEMINI_API_KEY=AIzaSyBRrf3oC4E0p9SgjLJg78AFfdWtRgVyqvE

# Forex Data APIs (optional)
ALPHA_VANTAGE_API_KEY=RC86UQHJGXT2X94Q
YAHOO_FINANCE_ENABLED=true

# Logging
LOG_LEVEL=INFO
```

6. Key Services & Components

- **WebSocket Manager:** Manages real-time WebSocket connections and message broadcasting

- **Forex Data Service:** Fetches and processes live forex market data

- **AI Analysis Service:** Performs AI-driven market analysis and predictions

- **ML Processor:** Handles machine learning model processing and inference

- **Trading Bot Service:** Automates trading decisions and order execution

- **Notification Service:** Sends real-time alerts and notifications to users

- **Connection Manager:** Manages database and service connections

7. Setup & Installation

Backend Setup:

1. Navigate to Backend directory: **cd Backend**

2. Create virtual environment: **python -m venv .venv**

3. Activate virtual environment: **.venv\Scripts\activate**

4. Install dependencies: **pip install -r requirements.txt**

5. Configure .env file with necessary API keys and settings

6. Run the application: **python run.py**

Frontend Setup:

1. Navigate to Frontend directory: **cd Frontend**

2. Install dependencies: **flutter pub get**

3. Configure API endpoints in the app configuration

4. Run on desired platform: **flutter run -d [device]**

8. Requirements & Dependencies

Python Packages (Backend):

```
#■ ■C■o■r■e■ ■B■a■c■k■e■n■d■ ■R■e■q■u■i■r■e■m■e■n■t■s■ ■(■M■i■n■i■m■a■l■)■
■
■f■a■s■t■a■p■i■=■=■0■.■1■0■9■.■0■
■
■u■v■i■c■o■r■n■[■s■t■a■n■d■a■r■d■]■=■=■0■.■2■7■.■0■
■
■w■e■b■s■o■c■k■e■t■s■=■=■1■2■.■0■
■
■p■y■t■h■o■n■-■m■u■l■t■i■p■a■r■t■=■=■0■.■0■.■6■
■
■
■
■#■ ■D■a■t■a■b■a■s■e■
■
■m■o■t■o■r■=■=■3■.■3■.■2■
■
■p■y■m■o■n■g■o■=■=■4■.■6■.■1■
■
■
■
■#■ ■D■a■t■a■ ■P■r■o■c■e■s■s■i■n■g■
■
■p■a■n■d■a■s■=■=■2■.■1■.■4■
■
■n■u■m■p■y■=■=■1■.■2■6■.■3■
■
■p■a■n■d■a■s■-■t■a■=■=■0■.■4■.■7■1■b■0■
■
■
■
■#■ ■H■T■T■P■ ■&■ ■U■t■i■l■i■t■i■e■s■
■
■h■t■t■p■x■=■=■0■.■2■6■.■0■
■
■r■e■q■u■e■s■t■s■=■=■2■.■3■1■.■0■
■
■p■y■t■h■o■n■-■d■o■t■e■n■v■=■=■1■.■0■.■0■
■
■p■y■d■a■n■t■i■c■=■=■2■.■5■.■2■
■
■p■y■d■a■n■t■i■c■-■s■e■t■t■i■n■g■s■=■=■2■.■1■.■0■
■
■p■y■t■h■o■n■-■d■a■t■e■u■t■i■l■=■=■2■.■8■.■2■
■
■
■
■#■ ■D■a■t■a■ ■S■o■u■r■c■e■s■
■
■y■f■i■n■a■n■c■e■=■=■0■.■2■.■3■3■
■
```

Flutter Packages (Frontend):

```
name: forex_companion
```

```yaml
description: AI-powered Forex trading assistant
publish_to: 'none'
version: 1.0.0+1

environment:
  sdk: '>=3.0.0 <4.0.0'

dependencies:
  flutter:
    sdk: flutter

  # UI
  cupertino_icons: ^1.0.8

  # State Management
  provider: ^6.1.1

  # HTTP & WebSocket
  http: ^1.2.0
  web_socket_channel: ^3.0.3

  # Date Formatting
  intl: ^0.20.2

  # Firebase
  firebase_core: ^4.3.0
  firebase_auth: ^6.1.3
  cloud_firestore: ^6.1.1
  firebase_storage: ^13.0.5

  # Gemini AI
  google_generative_ai: ^0.4.7

  # Local Storage
  shared_preferences: ^2.2.2
  sqflite: ^2.3.0
  path_provider: ^2.1.1

  # Utilities
  uuid: ^4.3.3
  flutter_dotenv: ^6.0.0

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^6.0.0

flutter:
  uses-material-design: true
```

Project Summary

TAJIR is a sophisticated trading platform combining real-time data processing, AI analysis, and user-friendly interfaces. The architecture supports scalability and real-time updates through WebSocket connections, while the multi-platform frontend ensures accessibility across all major platforms.

Documentation Generated: January 17, 2026 at 21:57:38