

Market Magician Deployment Manual

Introduction

The purpose of this document is to provide a step-by-step guide with all the requirements and prerequisites necessary for developers to deploy our application successfully

Prerequisites

Developers need the following software on their system

- Python 3.10
- NodeJS
- PostgreSQL

Deployment Steps

Clone Repository

The first step is to clone the repository into your working directory using a command line interface (CLI) such as bash, git bash, cmd, or powershell and the following git command:

```
git clone github_repo_url local_repo_name
```

Here *github_repo_name* is the HTTPS or SSH clone link that can be found at <https://github.com/htmw/2024F-Artificial-Asynchrony>.

For example, the following line clones the repository using the HTTPS link into a local repo named *project_repo*.

```
git clone https://github.com/htmw/2024F-Artificial-Asynchrony.git project_repo
```

Backend Setup

To setup the backed, the following assumptions are made:

1. A virtual environment using python 3.10 has been made
2. A PostgreSQL database and super user has been made

There are several different ways to create a virtual environment (venv, conda, virtualenv) with easily accessible tutorials. Use whichever method works best for you. The same goes for creating a PostgreSQL database and user.

The location of the virtual environment is up to the developer. It can even be placed in the same location as the local repository as long as it is **NOT** tracked in the repo.

Install Python Libraries

To begin, open a CLI instance in the *project_repo\market_magician_dev* directory

At this location activate the virtual environment. In a windows environment run the batch script.

```
\path_to_venv\activate.bat
```

If successful, the name of the virtual environment should appear before the CLI's prompt indicator.

With the virtual environment active and the CLI's working directory at *\market_magician_dev*, pip install the required libraries from the appropriate requirements.txt file. There are currently two, one for windows and other for mac (requirements_mac.txt).

```
pip install -r requirements.txt
```

If you cannot find the requirements.txt file, double check to ensure you are in the correct directory.

Set PostgreSQL Database Settings

Assuming the developer has created a PostgreSQL database, the developer needs to update the settings.py file located at *\market_magician_dev\market_magician*.

There is a block of code that looks like the following:

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql",
        "NAME": os.getenv("DB_NAME", "your_db_name"),
        "USER": os.getenv("DB_USER", "your_user_name"),
        "PASSWORD": os.getenv("DB_PASSWORD", "password"),
        "HOST": os.getenv("DB_HOST", "localhost"),
        "PORT": os.getenv("DB_PORT", "5432"), } }
```

Replace text in italics with your database details. The host and port are default values, but the developer can change them to suit their needs.

Migrate and Run

The last steps are to migrate the changes to the database, then run the server.

To migrate run the following in a CLI:

```
py manage.py migrate
```

Make sure the working directory is still pointing to `\market_magician_dev\`

If no errors occurred, the developer should now be ready to run the Django server.

To do so the following line is used:

```
py manage.py runserver
```

Frontend Setup

Assuming Node.js has been installed in the system, the front end set is more straightforward.

Open a CLI to `\market_magician_dev\frontend\`. You are in the right place if the `package.json` is in the working directory.

Here run the following:

```
npm install
```

This should install the necessary react packages into a `node_modules\` directory. Do **NOT** track this folder in the repository.

To run the front end server, run the following line

```
npm start
```

If any errors occur, check to make sure you are in the `\frontend\` directory.

