

Documentations

Table of Contents

README	2
Code Comments	3
Backend API Documentation	4

README

Our repository features detailed and instructive readme documents for both the frontend and backend components. These documents act as comprehensive guides, assisting future developers in effectively setting up, developing, and maintaining the project. They play a crucial role in preserving long-term code consistency and fostering robust development practices. For illustrative purposes, Figure 1 showcases excerpts from the readme files. To access the complete documentation, please visit our GitHub pages for the [main repository](#), [frontend](#), and [backend](#).

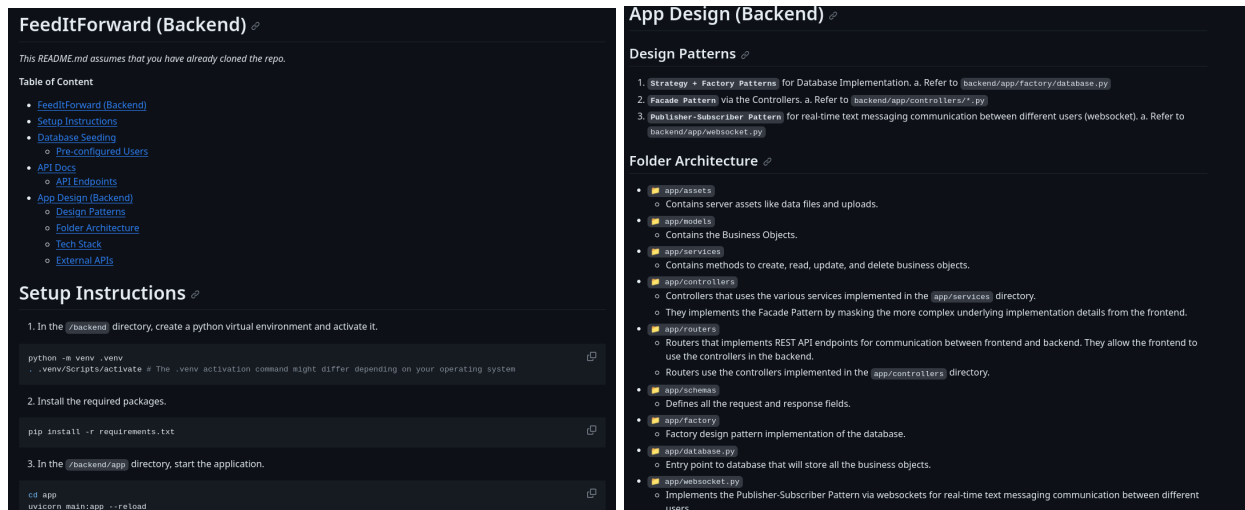


Figure 1: Excerpts of the README for the backend.

Code Comments

To enhance the readability and ease of understanding of our code, we have diligently incorporated comprehensive and insightful comments throughout. These annotations not only facilitate smoother collaboration among our development team members but also significantly aid new developers in familiarizing themselves with our codebase. To provide a clear example of our approach to code documentation, Figures 2 and 3 present selected excerpts from our well-commented code.

```
class FileController:
    # ----- File ----- #
    def uploadFile(user_id: int, file: UploadFile, db: Session):
        # Validate file type - Must be Image
        if file.content_type not in ["image/jpeg", "image/png", "image/gif"]:
            raise HTTPException(status_code=400, detail="Invalid file type. File type must be jpeg, png, or gif")

        # Create upload directory if it does not exist
        user_upload_directory = os.path.join(server_upload_base_directory, "user_{0}".format(user_id))
        if not os.path.exists(user_upload_directory):
            os.makedirs(user_upload_directory)

        # Destination File Path
        dest_file_path = os.path.join(user_upload_directory, file.filename)

        # Copy the file contents
        with open(dest_file_path, "wb") as buffer:
            shutil.copyfileobj(file.file, buffer)

        file_response = FileUploadResponse(
            file_path=str(os.path.join("user_{0}".format(user_id), file.filename))
        )

        return file_response
```

Figure 2: Excerpt of commented code

```
{isWeatherModalOpen && (
    <div className="absolute top-1/2 left-1/2 -translate-x-1/2 -translate-y-1/2 w-[75%] px-4 py-3 rounded bg-[#E3F6F5]">
        {/* Title */}
        <div className="flex flex-col items-center">
            <div className="font-bold text-[22px]">Weather Forecast</div>
            <div className="text-[14px] italic">
                Singapore | {date.toLocaleDateString()}
            </div>
        </div>

        {/* Weather Forecast */}
        {/* 24 Hour */}
        {weather24HrForecasts && (
            <div className="mt-4">
                <div className="font-bold underline text-[14px] text-center mb-1">
                    24 Hours
                </div>
                <div className="flex flex-col gap-1">
                    {REGIONS.map(region => (
                        <RegionWeatherForecast24HrRow
                            key={region}
                            weather24HrForecasts={weather24HrForecasts}
                            getWeatherIcon={getWeatherIcon}
                            region={region}
                        />
                    ))}
                </div>
            </div>
        )}
    </div>
)}

{/* 4 Day */}
{weather4DayForecasts && (
    <div className="mt-4">
        <div className="font-bold underline text-[14px] text-center mb-1">
```

Figure 3: Excerpt of commented code

Backend API Documentation

To enhance collaboration between our frontend and backend teams, we have developed an interactive backend API documentation using FastAPI. This documentation includes a user-friendly website that showcases various API routes, detailing the necessary input data, expected output data, and providing a feature to simulate request-response scenarios.

Figure 4 displays a screenshot of this website. Figure 5 displays a screenshot of a specific API endpoint, with expected request body and expected response body.

For detailed instructions on accessing and setting up the API documentation, please visit our GitHub page at [API Docs Setup Instructions](#).

**Note: A total of 100 API routes are documented.*

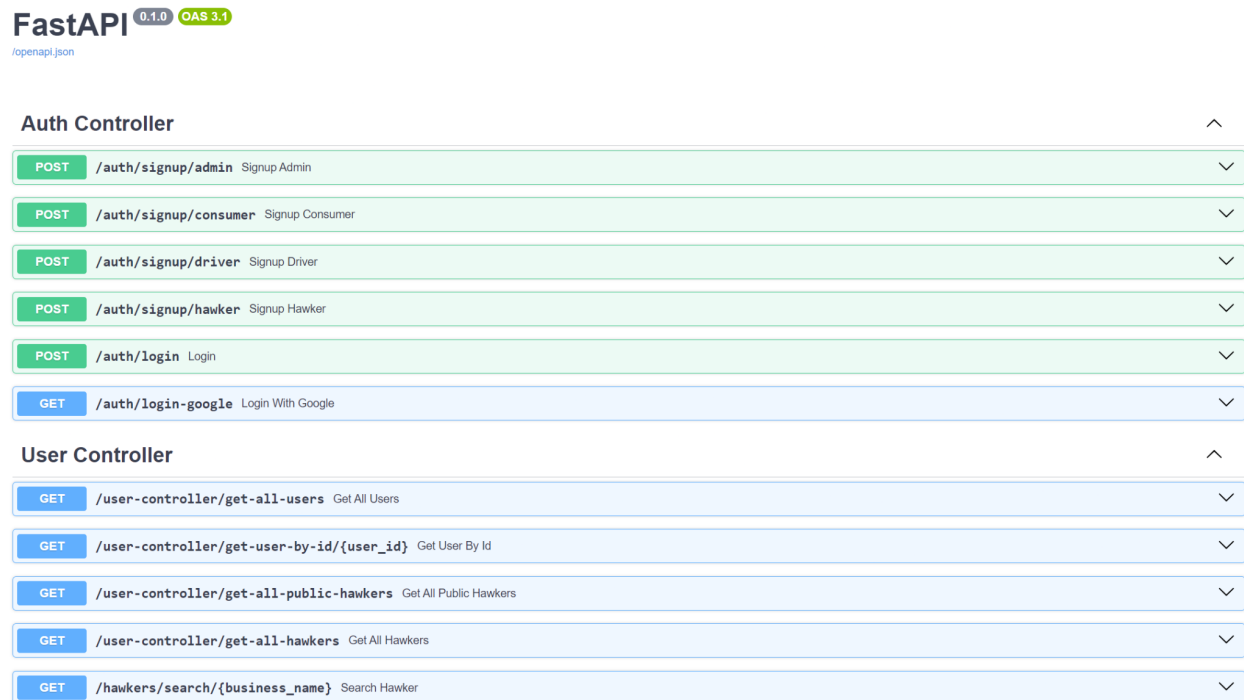


Figure 4: Screenshot of backend API docs website.

GET

/user-controller/get-all-hawkers

Get All Hawkers

⌵

Parameters

Try it out

Name	Description
skip integer (query)	Default value : 0 <input type="text" value="0"/>
limit integer (query)	Default value : 100 <input type="text" value="100"/>

Responses

Code	Description	Links
200	Successful Response <div>Media type <div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema</div><pre>{ "hawker_id": 0, "overall_rating": 0, "business_name": "string", "operating_hours": "string", "food_type": "string", "geometry": { "type": "Point", "latitude": 0, "longitude": 0 }, "is_registered": true, "user_id": 0, "user": { "name": "string", "email": "string", "contact_number": "string", "address": "string", "user_id": 0, "profile_picture": "", "role": "Admin", "ban": false } }</pre></div>	No links
422	Validation Error <div>Media type <div>application/json</div><div>Example Value Schema</div><pre>{ "detail": { { "loc": ["string", 0], "msg": "string", "type": "string" } } }</pre></div>	No links

Figure 5: Screenshot of a specific API endpoint, with expected request body and expected response body.