Dev Environment Setup:

This document is meant for developers in the project.  
Please follow below steps to get started with the project and this document will be updated as we progress on the project.

*1.* **Cloning GIT repo ->***git clone https://github.com/madhunica-balasubramanian/EmployeePerfMonitoringSystem.git /Users/madhunica/Desktop/CMPE272/PROJECT/development --progress*

*cd /Users/madhunica/Desktop/CMPE272/PROJECT/development*

*git checkout development*

*Please create a feature branch for whatever feature you are working on and push to dev branch only if your feature is working end-end.*

**2.** **Create Structure in VS Code for frontend using react.js , backend using python, database setup using postgres.**

***BackEnd***

*cd /Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSystem*

*mkdir -p backend/app/{models,routes,services,utils}*

*mkdir -p backend/tests*

*touch backend/app/\_\_init\_\_.py backend/app/config.py backend/app/main.py backend/requirements.txt*

*Structure*

*backend/*

*├── app/*

*│ ├── \_\_init\_\_.py*

*│ ├── models/*

*│ ├── routes/*

*│ ├── services/*

*│ └── utils/*

*├── tests/*

*├── config.py*

*├── requirements.txt*

*└── main.py*

***FrontEnd***

*mkdir -p frontend/src/{components,pages,services,utils}*

*touch frontend/src/{App.js,index.js}*

*mkdir frontend/public*

*touch frontend/package.json frontend/README.md*

*frontend/*

*├── public/*

*├── src/*

*│ ├── components/*

*│ ├── pages/*

*│ ├── services/*

*│ ├── utils/*

*│ ├── App.js*

*│ └── index.js*

*├── package.json*

*└── README.md*

***Database***

*mkdir -p database/migrations*

*touch database/init.sql*

*database/*

*├── migrations/*

*└── init.sql*

***Docker Setup***

*mkdir -p docker*

*touch docker/docker-compose.yml docker/Dockerfile.backend docker/Dockerfile.frontend*

*docker/*

*├── docker-compose.yml*

*├── Dockerfile.backend*

*└── Dockerfile.frontend*

touch .gitignore README.md , touch requirements.txt , cd backend , touch requirements.txt

**3.** **Setting up virtual environment and install dependencies**

*# Create and activate virtual environment*

python -m venv venv

source venv/bin/activate

pip install fastapi uvicorn psycopg2-binary sqlalchemy pydantic python-dotenv pytest

pip freeze > backend/requirements.txt

4. Running backend

Ensure you have the following lines in main.py (As its not abe to resolve app/ path)

***import sys***

***import os***

***sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_))))***

**Run main.py from the VS CODE.**

You should see something like this.

*/Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSystem/venv/lib/python3.11/site-packages/pydantic/\_internal/\_config.py:373: UserWarning: Valid config keys have changed in V2:*

*\* 'orm\_mode' has been renamed to 'from\_attributes'*

*warnings.warn(message, UserWarning)*

*INFO: Will watch for changes in these directories: ['/Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSystem/backend/app']*

*INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)*

*INFO: Started reloader process [19667] using StatReload*

*/Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSystem/venv/lib/python3.11/site-packages/pydantic/\_internal/\_config.py:373: UserWarning: Valid config keys have changed in V2:*

*\* 'orm\_mode' has been renamed to 'from\_attributes'*

*warnings.warn(message, UserWarning)*

*INFO: Started server process [19669]*

*INFO: Waiting for application startup.*

*INFO: Application startup complete.*

*/Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSyste*

*m/venv/bin/python /Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSystem/backend/app/main.py*

*In browser, http://127.0.0.1:8000/*

You can restart fastAPI server using,

*uvicorn app.main:app --reload*

5. Go to frontend and install required dependencies

# Navigate to frontend directory

cd frontend

Install npm install react-router-dom axios

# Initialize React app with Create React App

npx create-react-app .

# Install necessary dependencies

npm install axios react-router-dom @mui/material @mui/icons-material @emotion/react @emotion/styled recharts jwt-decode

**Navigate to FrontEnd and run the react App**

*cd frontend/*

*npm install*

*npm start dev*

*You can now view frontend in the browser*

*Local: http://localhost:3000*

*On Your Network: http://10.0.0.100:3000*

Note: if you encounter node.js issue. It typically occurs when you're using Node.js version 17 or later, which enforces stricter OpenSSL requirements.

If you don’t hve nvm installed,

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash

Always , do installations inside venv

rm -rf node\_modules package-lock.json

To remove and reinstall dependencies,

*rm -rf node\_modules package-lock.json*

*npm install*

Please follow the warnings and resolve any dependencies you have in VS code. The working set of package versions are as follows.

Once you are all set never delete package-lock.json

Let’s say you cloned or updated your repo.

*To reinstall* ***exact same versions****, run:*

*npm ci*

Always ensure, you commit these files.

*git add package.json package-lock.json*

*git commit -m "msg"*

To view current package version,

*npm list --depth=0*

*Sample snapshot,*

*(venv) madhunica@Madhunicas-MacBook-Air frontend % npm list --depth=0*

*frontend@0.1.0 /Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSystem/frontend*

*├── @emotion/react@11.14.0*

*├── @emotion/styled@11.14.0*

*├── @mui/icons-material@7.0.2*

*├── @mui/material@7.0.2*

*├── @testing-library/dom@10.4.0*

*├── @testing-library/jest-dom@6.6.3*

*├── @testing-library/react@16.3.0*

*├── @testing-library/user-event@13.5.0*

*├── axios@1.8.4*

*├── core-js@3.41.0*

*├── jwt-decode@4.0.0*

*├── react-dom@19.1.0*

*├── react-router-dom@7.5.0*

*├── react-scripts@5.0.1*

*├── react@19.1.0*

*├── recharts@2.15.2*

*├── web-vitals@2.1.4*

*├── webpack-cli@6.0.1*

*└── webpack@5.99.5*

*(venv) madhunica@Madhunicas-MacBook-Air frontend % node -v*

*v22.14.0*

*(venv) madhunica@Madhunicas-MacBook-Air frontend % npm -v*

*10.9.2*

*(venv) madhunica@Madhunicas-MacBook-Air frontend %*

*This enforces node.js version.*

*echo "22.14.0" > .nvmrc*

*nvm use*

6. Please follow below steps to work with postgres sql

In this file config.py, you will find the database name

*DATABASE\_URL = os.getenv("DATABASE\_URL", "postgresql://postgres:postgres@localhost:5432/wellness\_db")*

*Breaking this down,*

*postgresql:// -> DB type (PostgreSQL)*

*postgres:postgres -> username:password*

*localhost:5432 -> host:port*

*wellness\_db -> database name*

*Note that, All API endpoints will be prefixed with this (e.g., /api/v1/users, /api/v1/login, etc.)*

**Now, How do we create this database ?**

1. Install postgres if not installed already.
2. in mac, brew services start postgresql@15

*(venv) madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %*

*(venv) madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem % brew services restart postgresql@15*

*Stopping `postgresql@15`... (might take a while)*

*==> Successfully stopped `postgresql@15` (label: homebrew.mxcl.postgresql@15)*

*==> Successfully started `postgresql@15` (label: homebrew.mxcl.postgresql@15)*

1. In terminal, access via psql postgres (psql -U postgres -d wellness\_db)

*(venv) madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem % psql postgres*

*psql (15.12 (Homebrew))*

*Type "help" for help.*

*postgres=# CREATE DATABASE wellness\_db;*

*CREATE DATABASE*

1. Confirm whether the URL works,

*psql -U postgres -h localhost -p 5432 wellness\_db*

*If prompted for a password, enter postgres (the one in your config.py)*

1. Add all this logic in docker-compose.yaml , Please refer that file for better understanding.

**Other issues and potential ways to fix the problems**

1. After changing enum logic, always run alembic again . Please follow below steps if you modify any enum logic in models.py (table creation logic)

I. Inside your backend folder , pip install alembic

Ii. cd backend/ -> alembic init alembic

Iii. Modify .init file to have proper database uRL , look for this definition

DATABASE\_URL (postgres db url)

Iv. modify the env.py file with the below contents,

import sys

import os

sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_))))

from app.models.base import Base

# target\_metadata = mymodel.Base.metadata

target\_metadata = Base.metadata

V. Run alembic revision --autogenerate -m "Fix enum types"

You must see the following message,

INFO [alembic.autogenerate.compare] Detected removed index 'ix\_users\_email' on 'users'

INFO [alembic.autogenerate.compare] Detected removed index 'ix\_users\_id' on 'users'

INFO [alembic.autogenerate.compare] Detected removed index 'ix\_users\_username' on 'users'

INFO [alembic.autogenerate.compare] Detected removed table 'users'

INFO [alembic.autogenerate.compare] Detected removed index 'ix\_departments\_id' on 'departments'

INFO [alembic.autogenerate.compare] Detected removed table 'departments'

Generating

/Users/madhunica/Desktop/CMPE272/PROJECT/EmployeePerfMonitoringSystem/backend/alembic/versions/41d11f38af1f\_fix\_enum\_types.py ... done

Vi. alembic upgrade head

1. If you run into alembic issue, like this.

ERROR [alembic.util.messaging] Target database is not up to date.  
FAILED: Target database is not up to date.

Python3 reset\_db.py

Alembic stamp head

Python3 init\_db.py

High Level Design

1. **DataBase Design**

wellness\_db=# \dt

List of relations

Schema | Name | Type | Owner

--------+-------------------------+-------+----------

public | alembic\_version | table | postgres

public | departments | table | postgres

public | employee\_roles | table | postgres

public | metric\_definition\_roles | table | postgres

public | metric\_definitions | table | postgres

public | metric\_records | table | postgres

public | users | table | postgres

(7 rows)

**USERS TABLE:**

wellness\_db=# \d users

Table "public.users"

Column | Type | Collation | Nullable | Default

-----------------+------------------------+-----------+----------+-----------------------------------

id | integer | | not null | nextval('users\_id\_seq'::regclass)

username | character varying(50) | | not null |

email | character varying(100) | | not null |

hashed\_password | character varying(100) | | not null |

first\_name | character varying(50) | | |

last\_name | character varying(50) | | |

employee\_id | character varying(20) | | not null |

role | role\_type\_enum | | not null |

department\_role | department\_role\_enum | | not null |

department\_id | integer | | |

is\_active | boolean | | |

role\_id | integer | | |

Indexes:

"users\_pkey" PRIMARY KEY, btree (id)

"ix\_users\_email" UNIQUE, btree (email)

"ix\_users\_id" btree (id)

"ix\_users\_username" UNIQUE, btree (username)

"users\_employee\_id\_key" UNIQUE CONSTRAINT, btree (employee\_id)

Foreign-key constraints:

"users\_department\_id\_fkey" FOREIGN KEY (department\_id) REFERENCES departments(id)

"users\_role\_id\_fkey" FOREIGN KEY (role\_id) REFERENCES employee\_roles(role\_id)

Referenced by:

TABLE "metric\_records" CONSTRAINT "metric\_records\_user\_id\_fkey" FOREIGN KEY (user\_id) REFERENCES users(id)

wellness\_db=#

**DEPARTMENT TABLE:**

wellness\_db=# \d departments

Table "public.departments"

Column | Type | Collation | Nullable | Default

-------------+------------------------+-----------+----------+-----------------------------------------

id | integer | | not null | nextval('departments\_id\_seq'::regclass)

name | character varying(100) | | not null |

type | department\_type\_enum | | not null |

description | character varying(255) | | |

Indexes:

"departments\_pkey" PRIMARY KEY, btree (id)

"ix\_departments\_id" btree (id)

Referenced by:

TABLE "metric\_definitions" CONSTRAINT "metric\_definitions\_department\_id\_fkey" FOREIGN KEY (department\_id) REFERENCES departments(id)

TABLE "users" CONSTRAINT "users\_department\_id\_fkey" FOREIGN KEY (department\_id) REFERENCES departments(id)

wellness\_db=#

**METRIC\_RECORD TABLE:**

wellness\_db=# \d metric\_records

Table "public.metric\_records"

Column | Type | Collation | Nullable | Default

---------------+--------------------------+-----------+----------+--------------------------------------------

id | integer | | not null | nextval('metric\_records\_id\_seq'::regclass)

user\_id | integer | | not null |

metric\_id | integer | | not null |

metric\_type | metric\_type\_enum | | not null |

value\_numeric | double precision | | |

value\_text | text | | |

value\_json | text | | |

recorded\_at | timestamp with time zone | | |

notes | text | | |

Indexes:

"metric\_records\_pkey" PRIMARY KEY, btree (id)

"ix\_metric\_records\_id" btree (id)

Foreign-key constraints:

"metric\_records\_metric\_id\_fkey" FOREIGN KEY (metric\_id) REFERENCES metric\_definitions(id)

"metric\_records\_user\_id\_fkey" FOREIGN KEY (user\_id) REFERENCES users(id)

wellness\_db=#

**METRIC\_DEFINITIONS TABLE:**

wellness\_db=# \d metric\_definitions

Table "public.metric\_definitions"

Column | Type | Collation | Nullable | Default

----------------------------+------------------------+-----------+----------+------------------------------------------------

id | integer | | not null | nextval('metric\_definitions\_id\_seq'::regclass)

metric\_name | character varying(100) | | not null |

metric\_description | text | | |

metric\_type | metric\_type\_enum | | not null |

department\_id | integer | | |

unit | character varying(20) | | |

metric\_formula | text | | |

metric\_formula\_description | text | | |

is\_aggregated | boolean | | |

is\_numeric | boolean | | |

value | text | | |

Indexes:

"metric\_definitions\_pkey" PRIMARY KEY, btree (id)

"ix\_metric\_definitions\_id" btree (id)

Foreign-key constraints:

"metric\_definitions\_department\_id\_fkey" FOREIGN KEY (department\_id) REFERENCES departments(id)

Referenced by:

TABLE "metric\_definition\_roles" CONSTRAINT "metric\_definition\_roles\_metric\_id\_fkey" FOREIGN KEY (metric\_id) REFERENCES metric\_definitions(id)

TABLE "metric\_records" CONSTRAINT "metric\_records\_metric\_id\_fkey" FOREIGN KEY (metric\_id) REFERENCES metric\_definitions(id)

wellness\_db=#

**EMPLOYEE\_ROLES TABLE::**

wellness\_db=# \d employee\_roles

Table "public.employee\_roles"

Column | Type | Collation | Nullable | Default

------------------+------------------------+-----------+----------+-------------------------------------------------

role\_id | integer | | not null | nextval('employee\_roles\_role\_id\_seq'::regclass)

role\_name | character varying(100) | | not null |

role\_description | text | | |

Indexes:

"employee\_roles\_pkey" PRIMARY KEY, btree (role\_id)

"employee\_roles\_role\_name\_key" UNIQUE CONSTRAINT, btree (role\_name)

"ix\_employee\_roles\_role\_id" btree (role\_id)

Referenced by:

TABLE "metric\_definition\_roles" CONSTRAINT "metric\_definition\_roles\_role\_id\_fkey" FOREIGN KEY (role\_id) REFERENCES employee\_roles(role\_id)

TABLE "users" CONSTRAINT "users\_role\_id\_fkey" FOREIGN KEY (role\_id) REFERENCES employee\_roles(role\_id)

wellness\_db=#

wellness\_db=#

**METRIC\_DEFINITON\_ROLES TABLE:**

wellness\_db=# \d metric\_definition\_roles

Table "public.metric\_definition\_roles"

Column | Type | Collation | Nullable | Default

-----------+---------+-----------+----------+---------

metric\_id | integer | | not null |

role\_id | integer | | not null |

Indexes:

"metric\_definition\_roles\_pkey" PRIMARY KEY, btree (metric\_id, role\_id)

Foreign-key constraints:

"metric\_definition\_roles\_metric\_id\_fkey" FOREIGN KEY (metric\_id) REFERENCES metric\_definitions(id)

"metric\_definition\_roles\_role\_id\_fkey" FOREIGN KEY (role\_id) REFERENCES employee\_roles(role\_id)

wellness\_db=#

1. **Departments, Roles and their respective data taken into consideration**

**Performance metrics and Wellness Metrics of USPS department:**

1. **ROLE: Mail delivery Agent**

EmployeeId, Department, Role, and Date:  
 These fields serve as identifiers and metadata for each record.

Performance Metrics for Mail Delivery Agent:

* parcelsDeliveredOnTime: Number of parcels delivered on time for the workday.
* parcelsDeliveredLate: Number delivered late.
* parcelsUndelivered: Number that were undelivered.
* redeliveryAttempts: How many times a redelivery was attempted.

Wellness Metrics for Mail Delivery Agent:

* distanceCovered: Total distance (in miles) covered during the workday.
* injuryReport: Any injury observed (here, a string; could be “None” or a brief description).
* weatherExposure: How severe the weather exposure was during work.
* sickDays: Number of sick days taken.
* stressLevel: A subjective rating (e.g., on a scale from 1 to 10).
* JobSatisfaction: scale (1-10)

As a supervisor, I cant edit any fields for mail delivery agent I can only view.

1. **ROLE: Front Office Admin**

employeeId, department, role, date:  
Basic metadata to identify the record and its context.

Performance Metrics for an Office Worker:

* 1. inquiriesHandled: The number of customer inquiries or calls handled during the workday.
  2. documentsProcessed: How many documents (e.g., forms, reports) were processed.
  3. errorsIdentified: The number of processing errors found during the day.
  4. complaintsResolved: Count of customer complaints successfully resolved.

Wellness Metrics for an Office Worker:

* 1. hoursSitting: Total hours spent sitting at a desk   
     (calculate from login - logout time - internal )
  2. breaksTaken: How many breaks the worker took during the shift.
  3. stressLevel: A subjective metric (e.g., on a scale of 1 to 10) indicating the work-related stress experienced.
  4. sickDays: Number of sick days taken (typically 0 on a single workday record).

As s supervisor, I can update customer feedback for front office admin and view the metrics but other fields are not editable.

**Performance metrics and Wellness Metrics of HealthCare department:**

1. **ROLE: Public Health Nurse**

employeeId, department, role, date:  
Basic metadata to identify the record and its context.

Performance data for Public Health Nurse

1. appointmentsScheduled: Total number of patient appointments set up
2. Patients Seen
3. Vaccinations Administered:

Wellness data for Public Health Nurse

1. hoursWorked: Total hours worked for the day
2. NightShifts for the day: Y/N
3. Sleep Hours: Sleep hours
4. stressLevel: Subjective stress rating (e.g., on a scale from 1–10).
5. sickDays: Typically 0 for the day if the employee is present.
6. **ROLE: Medical Admin Assistant**

employeeId, department, role, date:  
Basic metadata to identify the record and its context.

Performance Metrics for Public Health Nurse

1. recordsProcessed: Number of administrative records or documents processed.
2. inquiriesHandled: Count of telephone or email inquiries handled.
3. errorsInDataEntry: Number of errors recorded during data entry.
4. CallResponseTime:
5. NumberORecordsCorrected:

Wellness Metrics for Public Health Nurse

1. hoursWorked: Total hours worked for the day.
2. NightShifts for the day: Y/N
3. Sleep Hours: Sleep hours.
4. stressLevel: Subjective stress rating (e.g., on a scale from 1–10).
5. sickDays: Typically 0 for the day if the employee is present.

As a supervisor, I can update customer feedback for the Medical Admin assistant.

**USPS**

-- Performance Metrics

on\_time\_delivery\_rate DECIMAL(5,2), -- percentage

customer\_feedback TEXT, -- complaints/praise (free text or enum)

delivery\_accuracy DECIMAL(5,2), -- percentage

-- Wellness Metrics

physical\_activity\_steps INTEGER,

physical\_activity\_miles DECIMAL(5,2),

injury\_report TEXT,

risk\_level TEXT, -- e.g., 'LOW', 'MEDIUM', 'HIGH'

weather\_exposure TEXT, -- e.g., 'rain', 'extreme heat', etc.

**Healthcare -> Metric Definitions**

-- Performance Metrics

patients\_attended INTEGER,

patient\_satisfaction\_score DECIMAL(3,2), -- scale: 1.00 to 5.00

report\_timeliness\_minutes DECIMAL(5,2) -- time to report/resolve

–Wellness Metrics

burnout\_risk\_index DECIMAL(3,2), -- scale: 0.00 to 1.00

self\_assessment\_survey JSONB, -- structured responses (e.g., { "q1": 5, "q2": 3 })

sleep\_hours DECIMAL(4,2),

stress\_level TEXT -- e.g., 'LOW', 'MODERATE', 'HIGH'

**Transportation -> Metric Definitions**

–Performance Metrics

project\_milestones\_completed INTEGER,

maintenance\_check\_timeliness DECIMAL(5,2), -- hours

incident\_free\_days INTEGER

–Wellness Metrics

sleep\_tracking\_hours DECIMAL(4,2),

fatigue\_risk\_index DECIMAL(3,2), -- 0.00 to 1.00

shift\_duration\_hours DECIMAL(4,2)

Manual form to submit daily/weekly inputs

Auto-fill certain defaults (like shift duration, step goal, etc.)

→ Decide according to each department what data can be filled manually by the supervisor and what data can be submitted by the employee and comes for approval to manager.

Both projects use **JWT**. So you just need to ensure your login/register routes match the expected structure (e.g., POST /api/v1/login/access-token).

)

FrontEnd Design

**FRONTEND DESIGN**

1. Landing page -> login page

* No User Registration option in Landing login Page
* Login as Supervisor or Employee
* **Login as Supervisor,**
  + I don’t have performance or wellness metrics . No dashboard for that.
  + I can add user(ie., employee information)

{

"username": "patrick",

"email": "patrick@example.com",

"hashed\_password": get\_password\_hash("patrick123"),

"first\_name": "Patrick",

"last\_name": "Smith",

"role": RoleType.EMPLOYEE, (dropdown)

"department\_role": DepartmentRoleType.USPS\_MAIL\_CARRIER, (dropdown)

"department\_id": 1,

"employee\_id": "EMP004", (auto generated)

"is\_active": True

},

When existing employee added - throw error .

* + I can search for employees based on employee ID who only belongs to my department.(say I am the supervisor of USPS, I can only search employees from this department, other searches will not yield results)
  + I can view their performance and wellness metrics of an employee.
  + I can update a few fields in the performance and wellness data of an employee.(refer below section highlighted in yellow for the fields that a supervisor can edit in performance/wellness).
  + I can view performance and wellness metrics (aggregate) of my department entirely

- Login as Employee,

- I can view only my details

- I can view my personal information such as EmployeeID, FirstName, LastName, department Role (Post Office worker, Front Desk Admin, Health care nurse etc), Department name (USPS or HealthCare). ( None of the fields are editable) - Just View.

- I can record my performance data for the day.

Say, I login on 23rd april,2025 5:00PM to this portal, I can record my performance data . Please have below fields displayed as per the employee’s role and department.  
Supervisor - Manager and otherwise we have 2 employee roles defined per department as of now.

class DepartmentType(str, enum.Enum):

USPS = "USPS"

HEALTHCARE = "HEALTHCARE"

class RoleType(str, enum.Enum):

EMPLOYEE = "EMPLOYEE"

SUPERVISOR = "SUPERVISOR"

ADMIN = "ADMIN"

# Healthcare roles

HEALTHCARE\_SUPERVISOR = "HEALTHCARE\_SUPERVISOR"

HEALTHCARE\_ADMIN = "HEALTHCARE\_ADMIN"

HEALTHCARE\_NURSE = "HEALTHCARE\_NURSE"

# USPS roles

USPS\_SUPERVISOR = "USPS\_SUPERVISOR"

USPS\_MAIL\_CARRIER = "USPS\_MAIL\_CARRIER"

USPS\_OFFICE\_ADMIN = "USPS\_OFFICE\_ADMIN"

I. Department : USPS, Role: USPS\_MAIL\_CARRIER Employee

Performance Data in the form to be displayed(to edit):

* parcelsDeliveredOnTime: Number of parcels delivered on time for the workday - TextField (only Integers allowed)
* parcelsDeliveredLate: Number delivered late - TextField (only integers allowed)
* parcelsUndelivered: Number that were undelivered - TextField (only integers allowed)
* redeliveryAttempts: How many times a redelivery was attempted - TextField (only integers allowed)

Wellness Data in the form to be displayed (to edit):

* distanceCovered: Total distance (in miles) covered during the workday - TextField (in miles)
* injuryReport: Any injury observed (here, a string; could be “None” or a brief description) - Option (yes/ no) -> If Yes , generate a small text box to type about the description of injury.
* weatherExposure: How severe the weather exposure was during work - Dropdown box - Give three options (rain, extreme heat, extreme cold)
* sickDays Taken for the Month: Number of sick days taken - TextField - allow values between 1 - 30
* stressLevel: A subjective rating (e.g., on a scale from 1 to 10) - Scale to choose from (1-10)
* JobSatisfaction: scale to choose from (1-10)

II. Department : USPS, Role: USPS\_OFFICE\_ADMIN

Performance Data in the form to be displayed(to edit by employee):

* + Customer Tickets Opened: The number of customer service tickets received for the day . - Textfield.
  + Customer Tickets Handled and Resolved: Number of customer service tickets handled and moved to resolved state for the day.
  + **Customer Satisfaction Score :** This is the data that will be updated by supervisor - Have a dropdown rating of 1-5 (5 being highest). Keep it disabled for employee.
  + **Customer Feedback :** All the feedback given by customers for the employee. Keep a big text box so that supervisor can update this. Keep it disabled for employee.
  + documentsProcessed: Number of documents processed for the day - TextField - Only Integers allowed.

Wellness Data in the form to be displayed (to edit by employee):

* sickDays Taken for the Month: Number of sick days taken - TextField - allow values between 1 - 30
* hoursSitting: Total number of hours spent during the day - Dropdown box ( 5 , 6, 7, 8 and > 8 hours).
* stressLevel: A subjective rating (e.g., on a scale from 1 to 10) - Scale to choose from (1-10)
* JobSatisfaction: scale to choose from (1-10)

III. Department :HEALTHCARE, Role: HEALTHCARE\_ADMIN

Performance Data to be displayed(to edit by employee):

* PatientsRecordsProcessed: Total number of patient records processed and data entry to system TextField
* Patients Call Inquiries Handled: Number of Patient call inquires attended for the day - TextField
* NumberOfRecords returned for re-review or Erroneous records: - TextField
* **CallResponseTime : T**his field will be updated by supervisor. Keep a drop down box rating (1-5) and keep disabled for employee login

Wellness Data to be displayed (to edit by employee):

* NightShift Worked : Y/N option
* NumberofHours Worked: DropDownBox ( 5 , 6, 7, 8 and > 8 hours).
* stressLevel: A subjective rating (e.g., on a scale from 1 to 10) - Scale to choose from (1-10)
* JobSatisfaction: scale to choose from (1-10)

IV.. Department :HEALTHCARE, Role: HEALTHCARE\_NURSE

Performance Data to be displayed(to edit by employee):

* appointmentsScheduled: Total number of patient appointments set up- TextField
* Patients Seen: Number of Patients attended for the day - TextField
* Vaccinations Administered: Number of vaccinations I administered - TextField
* **PatientFeedback :** This is the aggregated feedback that will be updated by supervisor. Keep a big text box and keep disabled for employee login
* **Patient Satisfaction Score :** This will also be updated by supervisor, keep a dropdown box rating (1-5) and keep it disabled for the employee.

Wellness Data to be displayed (to edit by employee):

* NightShift Worked : Y/N option
* NumberofHours Worked: DropDownBox ( 5 , 6, 7, 8 and > 8 hours).
* stressLevel: A subjective rating (e.g., on a scale from 1 to 10) - Scale to choose from (1-10)
* JobSatisfaction: scale to choose from (1-10)
* As an employee, I can view my performance and wellness metrics (aggregated view only for the year - no option to view by date / month / year)

BackEnd Design

1. **Design of Backend**

backend/

├── app/

│ ├── auth/

│ │ ├── deps.py # Authentication-related dependencies (e.g., `get\_current\_user`, `get\_current\_user\_role`)

│ ├── models/

│ │ ├── models.py # SQLAlchemy models for database tables

│ │ ├── base.py # Base class for SQLAlchemy models and session setup

│ ├── routes/

│ │ ├── auth.py # Authentication routes (e.g., login)

│ │ ├── users.py # User-related routes (e.g., CRUD operations for users)

│ │ ├── metrics.py # Metric-related routes (e.g., fetching performance/wellness)

│ │ ├── metric\_records.py # Metric record-related routes for employees

│ ├── utils/

│ │ ├── security.py # Utility functions for security (e.g., password hashing, JWT creation)

│ ├── config.py # Configuration file (e.g., database URL, secret keys)

│ ├── main.py # Entry point for the FastAPI application

├── init\_db.py # Database initialization and seeding logic

├── metric\_definitions.json # JSON file for seeding metric definitions

#### **Authentication Module**

* File: auth/deps.py
* Purpose:
  + Provides dependencies for authentication, such as get\_current\_user and get\_current\_user\_role.
  + Handles JWT decoding and validation.
* Example Functions:
  + get\_current\_user: Extracts the current user from the JWT token.
  + get\_current\_user\_role: Extracts the user's role from the JWT token.

#### **Models**

* File: models/models.py
* Purpose:
  + Defines the database schema using SQLAlchemy models.
  + Includes relationships between tables (e.g., User, Department, MetricDefinition).
* Example Models:
  + User: Represents users in the system (e.g., employees, supervisors, admins).
  + Department: Represents departments (e.g., USPS, Healthcare).
  + MetricDefinition: Represents performance and wellness metrics.

#### **Routes**

* Folder: routes/
* Purpose:
  + Contains FastAPI route definitions for different functionalities.
* Files:
  + auth.py: Handles login and authentication-related routes.
  + users.py: Provides CRUD operations for users and role-based access control.
  + metrics.py: Handles fetching and updating metrics (e.g., performance, wellness).
  + metric\_records.py: Handles metric record-related operations.

#### **Utilities**

* Folder: utils/
* Purpose:
  + Contains helper functions for common tasks.
* File: security.py
  + Purpose: Provides security-related utilities.
  + Example Functions:
    - get\_password\_hash: Hashes passwords using a secure algorithm.
    - create\_access\_token: Creates JWT tokens for authentication.

#### **Database Initialization and Seeding**

* File: init\_db.py
* Purpose:
  + Initializes the database schema.
  + Seeds initial data into the database (e.g., departments, users, metrics).
* Example Functions:
  + init\_db: Creates database tables.
  + seed\_departments: Seeds department data.
  + seed\_admin\_user: Seeds an admin user.
  + seed\_supervisor\_user: Seeds supervisor users.
  + seed\_employee\_user: Seeds employee users.
  + seed\_metric\_definitions: Seeds metric definitions from metric\_definitions.json.

#### **Configuration**

* File: config.py
* Purpose:
  + Stores configuration settings (e.g., database URL, secret keys).
  + Centralizes environment-specific settings.

#### **Main Application**

* File: main.py
* Purpose:
  + Entry point for the FastAPI application.
  + Includes middleware, route registration, and startup tasks.
* Key Features:
  + Registers routers from routes/.
  + Seeds initial data during application startup using @app.on\_event("startup").

### **Dependencies Between Components**

#### **Database**

* Models (models/models.py) depend on:
  + base.py for the SQLAlchemy Base class and session setup.
* Seeding (init\_db.py) depends on:
  + Models (models/models.py) for inserting data into tables.
  + Security utilities (utils/security.py) for hashing passwords.

#### **Authentication**

* Routes (auth.py) depend on:
  + auth/deps.py for authentication dependencies.
  + utils/security.py for creating JWT tokens.

#### **Routes**

* Routes (users.py, metrics.py, etc.) depend on:
  + Models (models/models.py) for database queries.
  + Authentication dependencies (auth/deps.py) for role-based access control.

### **Code Flows**

#### **Login and Authentication**

1. Startup:
   * main.py initializes the application and seeds initial data.
2. Authentication:
   * Users log in via auth.py, and their role is included in the JWT token.
3. Role-Based Access:
   * Routes use get\_current\_user\_role to enforce role-based permissions.
4. Database:
   * Models define the schema, and init\_db.py populates the database with initial data.

## **✅ Should Metric Submission Be Date-Specific or Date-Range?**

### **🔹 Recommendation: Date-Specific Submissions (per day)**

### **Why?**

1. **Daily granularity** is easier to track, audit, and visualize (e.g., charts, trends).
2. Avoids ambiguity — wellness and performance are **usually evaluated daily** (e.g., parcels delivered *today*).
3. Makes enforcing one submission per day simple.

💡 You can always **aggregate later** (weekly, monthly), but daily data provides maximum flexibility and traceability.

### **Assumptions/WorkArounds**

For testing purpose, following data are preseeded now

Refer init\_db.py where supervisor, department, netric\_Definitions and users data are pre populated.

After backend and frontend are in place,

Remove

employees data (Supervisor can add employees via frontend)

Keep

Admin, supervisor, department and metric\_Definitons data (No front end provision to add or modify these)

Testcases - Screenshots and logs

Run your program::

**Backend:**

**python -m venv venv**

**source venv/bin/activate**

**export SECRET\_KEY=”Afoxfellintothewellandcriedforhelpbutnoonehelped”**

**uvicorn app.main:app --reload**

**run main.py**

**psql -U postgres -d wellness\_db**

**Frontend:**

**Cd frontend/**

**npm install**

**npm run dev**

**npm cache clean --force**

**Clean up : rm package-lock.json , rm -rf node\_modules**

**Upgrade node.js to version 20**

**Install typescript**

**npm install --save-dev typescript @types/react @types/node**

**npm install @radix-ui/react-slot**

**Npm install class**

Now using node v20.19.1 (npm v10.8.2)

madhunica@Madhunicas-MacBook-Air frontend % nvm use 20

Now using node v20.19.1 (npm v10.8.2)

madhunica@Madhunicas-MacBook-Air frontend % npm install

added 10 packages, and audited 48 packages in 2s

11 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

madhunica@Madhunicas-MacBook-Air frontend %

madhunica@Madhunicas-MacBook-Air frontend % node -v

v20.19.1

1. **Role and auth module - Owner:Madhunica**

**Login as employee**

TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-d "username=jason&password=jason123" | jq -r .access\_token)

echo $TOKENVAL

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ1NzIzNTIzfQ.h3rGCRo0-p8pe6itqGeU5wL7Tp4dK5HYRnUle2zfGSk

TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-d "username=patrick&password=patrick123" | jq -r .access\_token)

**Login as Supervisor**

TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-d "username=jason&password=jason123" | jq -r .access\_token)

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem % echo $TOKENVAL

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ1Nzk4MDE5fQ.WMMuyTBpHXQ9DSoiaz3NW8UU49bJI00wypl-VjYULKY

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

**Add an employee to my department**

curl -X POST http://localhost:8000/api/v1/users/ \

-H "Content-Type: application/json" \

-H "Authorization: Bearer $TOKENVAL" \

-d '{

"username": "tom",

"email": "tom@example.com",

"password": "Tom!123",

"first\_name": "Tom",

"last\_name": "Cook",

"role": "EMPLOYEE",

"department\_role":"USPS\_MAIL\_CARRIER",

"department\_id": 1,

"is\_active": true

}'

{"username":"tom","email":"tom@example.com","first\_name":"Tom","last\_name":"Cook","role":"EMPLOYEE","department\_id":1,"is\_active":true,"id":8,"created\_at":null}% madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

**Add an employee to a different department - Not Allowed**   
curl -X POST http://localhost:8000/api/v1/users/ \

-H "Content-Type: application/json" \

-H "Authorization: Bearer $TOKENVAL" \

-d '{

"username": "tom",

"email": "tom@example.com",

"password": "Tom!123",

"first\_name": "Tom",

"last\_name": "Cook",

"role": "EMPLOYEE",

"department\_role":"USPS\_MAIL\_CARRIER",

"department\_id": 2,

"is\_active": true

}'

{"detail":"Supervisors can only add employees to their own department."}**%** madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

**View my department employees as a supervisor**

**View a particular employee’s profile detail as a supervisor**

**Delete an employee detail from my department**

**Delete an employee detail from other department**

**Login as Admin**TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-d "username=admin&password=adminpassword123" | jq -r .access\_token)

TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-d "username=admin&password=adminpassword123" | jq -r .access\_token)

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem % echo $TOKENVAL

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJhZG1pbiIsInJvbGUiOiJBRE1JTiIsInVzZXJfaWQiOjEsImV4cCI6MTc0NTc5NTEzN30.YbXOpm\_22WVaEO7PLirISHDqgZR0QAbr4rb7Jmpreuw

**View Supervisors - As Admin**

curl -X GET "http://localhost:8000/api/v1/users/supervisors" \

-H "Authorization: Bearer $TOKENVAL" \

-H "accept: application/json"

[{"username":"johnny","email":"johnny@example.com","first\_name":"Johnny","last\_name":"Doe","role":"SUPERVISOR","department\_id":2,"is\_active":true,"id":3,"created\_at":null},{"username":"jason","email":"jason@example.com","first\_name":"Jason","last\_name":"Smith","role":"SUPERVISOR","department\_id":1,"is\_active":true,"id":9,"created\_at":null}]% madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

**Add supervisor - as admin**curl -X POST http://localhost:8000/api/v1/users/ \

-H "Content-Type: application/json" \

-H "Authorization: Bearer $TOKENVAL" \

-d '{

"username": "supervisor3",

"email": "supervisor3@example.com",

"password": "Supervisor!123",

"first\_name": "Johnnie",

"last\_name": "Doey",

"role": "SUPERVISOR",

"department\_role":"USPS\_SUPERVISOR",

"department\_id": 1,

"is\_active": true

}'

{"username":"supervisor3","email":"supervisor3@example.com","first\_name":"Johnnie","last\_name":"Doey","role":"SUPERVISOR","department\_id":1,"is\_active":true,"id":9,"created\_at":null}%

**Delete supervisor - as admin**

curl -X DELETE http://127.0.0.1:8000/api/v1/users/supervisors/8 \

-H "Authorization: Bearer $TOKENVAL"

curl -X DELETE http://127.0.0.1:8000/api/v1/users/supervisors/2 \

-H "Authorization: Bearer $TOKENVAL"

wellness\_db=# SELECT \*from users;

id | username | email | hashed\_password | first\_name | last\_name | employee\_id | role | department\_role | department\_id | is\_active | role\_id

----+----------+---------------------+--------------------------------------------------------------+------------+-----------+-------------+------------+-----------------------+---------------+-----------+---------

1 | admin | admin@example.com | $2b$12$e6xxYh5seC/fR8Usw04SyOcvwJMLdUJNgWwYYxd19QX2jLvLs2pvG | System | Admin | EMP001 | ADMIN | ADMIN2 | | t |

3 | johnny | johnny@example.com | $2b$12$Pls/tU8.lBVqOtIBXOGuAeCFhJiepej34JkOOQ2lj5e8VlP/dsrTO | Johnny | Doe | EMP008 | SUPERVISOR | HEALTHCARE\_SUPERVISOR | 2 | t |

4 | patrick | patrick@example.com | $2b$12$dZhsYtKgcuO5mqwWU685Y.6ubAoW9AZEkOS24QwCPnW/.DRWP1cYy | Patrick | Smith | EMP004 | EMPLOYEE | USPS\_MAIL\_CARRIER | 1 | t |

5 | Robert | robert@example.com | $2b$12$NaheGQBw6r7/LJdyx1/6h.m3B9aVi6byVMqni7EtMrYY.YznwyNLy | Robert | Ross | EMP007 | EMPLOYEE | USPS\_OFFICE\_ADMIN | 1 | t |

6 | jane | jane@example.com | $2b$12$snEY/8MG1J535tSF52c85eB3rie1g1ETcjjB7JhR3Hn6gDVzalnLa | Jane | Doe | EMP005 | EMPLOYEE | HEALTHCARE\_NURSE | 2 | t |

7 | michael | michael@example.com | $2b$12$K7CvhxH.pPmZHd.BfXxV.emA9BcKLsnwQoPihy4svyjz1hhSVi2Tq | Michael | Johnson | EMP006 | EMPLOYEE | HEALTHCARE\_ADMIN | 2 | t |

(6 rows)

**Update supervisor - as admin**

curl -X PUT http://127.0.0.1:8000/api/v1/users/supervisors/2 \

-H "Authorization: Bearer $TOKENVAL" \

-H "Content-Type: application/json" \

-d '{

"username": "jason",

"email": "jason@example.com",

"first\_name": "Jason",

"last\_name": "Smither",

"role": "SUPERVISOR",

"department\_id": 1,

"department\_role": "USPS\_SUPERVISOR"

}'

{"username":"jason","email":"jason@example.com","first\_name":"Jason","last\_name":"Smither","role":"SUPERVISOR","department\_id":1,"is\_active":true,"id":2,"created\_at":null}% madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

2 | jason | jason@example.com | $2b$12$xhSVkbKBdPpKJ4OUafDwHOZnd39yq5N8XK2NcQAjJdG1LYlhorkba | Jason | Smither | EMP002 | SUPERVISOR | USPS\_SUPERVISOR | 1 | t |

(8 rows)

**View department as Admin**

curl -X GET "http://localhost:8000/api/v1/users/departments" \

-H "Authorization: Bearer $TOKENVAL" \

-H "accept: application/json"

[{"id":1,"name":"US Postal Service","type":"USPS","description":"Handles mail delivery"},{"id":2,"name":"Healthcare","type":"HEALTHCARE","description":"Federal health services"}]%

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

**Add department as Admin - Not Implemented - Facing alembic version issue (if time permits, will fix this later)**

1. **Perf and wellness module - Owner:Namita**

**uvicorn app.main:app --reload**

**Run main.py**

These are the major test cases that has to work and below you will see sub test cases that are required for these to work.

Test1: test login of admin (using pre seeded value)

Test2: test login of supervisor (using preseeded value)

Test3: test login of employees (1) (using preseeded value)

* add performance data for myself - employee 1,2,3,4
* add/update wellness data for myself - employee 1,2,3,4
* update performance data of employee 1,2,3,4
* update wellness data of employee 1,2,3,4
* view performance metric of employee
* view wellness metric of employee

Test4: test login of supervisor

* Search employee by ID.
* View performance metrics of that employee
* View wellness metrics of that employee
* view performance metric of my department
* view wellness metric of my department
* Edit certain performance metrics of an employee

1. **Valid case - Fetch all users from the table.**

curl -X GET <http://localhost:8000/api/v1/users/>

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem % curl -X GET http://localhost:8000/api/v1/users/

[{"username":"admin","email":"admin@example.com","first\_name":"System","last\_name":"Admin","role":"ADMIN","department\_id":null,"is\_active":true,"id":1,"created\_at":null},{"username":"jason","email":"jason@example.com","first\_name":"Jason","last\_name":"Smith","role":"SUPERVISOR","department\_id":1,"is\_active":true,"id":2,"created\_at":null},{"username":"johnny","email":"johnny@example.com","first\_name":"Johnny","last\_name":"Doe","role":"SUPERVISOR","department\_id":2,"is\_active":true,"id":3,"created\_at":null},{"username":"patrick","email":"patrick@example.com","first\_name":"Patrick","last\_name":"Smith","role":"EMPLOYEE","department\_id":1,"is\_active":true,"id":4,"created\_at":null},{"username":"Robert","email":"robert@example.com","first\_name":"Robert","last\_name":"Ross","role":"EMPLOYEE","department\_id":1,"is\_active":true,"id":5,"created\_at":null},{"username":"jane","email":"jane@example.com","first\_name":"Jane","last\_name":"Doe","role":"EMPLOYEE","department\_id":2,"is\_active":true,"id":6,"created\_at":null},{"username":"michael","email":"michael@example.com","first\_name":"Michael","last\_name":"Johnson","role":"EMPLOYEE","department\_id":2,"is\_active":true,"id":7,"created\_at":null}]**%** madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

1. **Valid case - Fetch only employee by id**

curl -X GET "http://localhost:8000/api/v1/users/2" -H "accept: application/json"

{"username":"jason","email":"jason@example.com","first\_name":"Jason","last\_name":"Smith","role":"SUPERVISOR","department\_id":1,"is\_active":true,"id":2,"created\_at":null}%

1. **Invalid case - Fetch non existent user by id**

curl -X GET "http://localhost:8000/api/v1/users/99" -H "accept: application/json"

{"detail":"User not found"}**%** madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem %

1. **Valid case - Fetch the logged in employee’s perfromance metrics**

Endpoint tested - @router.get("/performance-metrics")

Get performance metrics

1. **Valid case - Fetch the logged in employee’s wellness metrics**
2. **Valid case - Fetch the perfromance metrics of a particular employee - by supervisor**
3. **Valid case - Fetch the wellness metrics of a particular employee - by supervisor**
4. **Submit performance data of a logged in employee**
5. **Submit wellness data of a logged in employee**

1. **Delete employee by the supervisor - Namitha**

Curl command test

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl.exe -X DELETE "http://127.0.0.1:8000/api/v1/users/delete\_employee/5" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqb2hubnkiLCJyb2xlIjoiU1VQRVJWSVNPUiIsInVzZXJfaWQiOjMsImV4cCI6MTc0NTYzOTA5OH0.2NliCHQ50GbVFbvUvhG-1DOmNU-7p0Sor8nvHJWmEn8"

{"message":"Employee Robert deleted successfully."}

1. **Add employee by the supervisor - Namitha**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl.exe -X POST "http://127.0.0.1:8000/api/v1/users/create\_employee" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqb2hubnkiLCJyb2xlIjoiU1VQRVJWSVNPUiIsInVzZXJfaWQiOjMsImV4cCI6MTc0NTYzOTA5OH0.2NliCHQ50GbVFbvUvhG-1DOmNU-7p0Sor8nvHJWmEn8" ^

More? -H "Content-Type: application/json" ^

More? -d "{\"username\":\"Sheldon\",\"email\":\"sheldon@example.com\",\"password\":\"Sheldon@123\",\"first\_name\":\"Sheldon\",\"last\_name\":\"Cooper\",\"department\_role\":\"USPS\_MAIL\_CARRIER\",\"department\_id\":1,\"is\_active\":true}"

{"message":"Employee Sheldon created successfully!","employee\_id":"EMP009"}

1. **Login as employe**
2. **Login as Admin**
3. **Login as supervisor**
4. **Delete supervisor as Admin - Namitha**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl.exe -X DELETE "http://127.0.0.1:8000/api/v1/users/delete\_supervisor/3" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJhZG1pbiIsInJvbGUiOiJBRE1JTiIsInVzZXJfaWQiOjEsImV4cCI6MTc0NTY0MTQ2N30.MD1x49muDB1OVCWaMmFFtJj\_NyqPHv3QmlcbSWYnr14"

{"message":"Supervisor johnny deleted successfully."}

1. **Add supervisor as Admin - Namitha**

curl.exe -X POST "http://127.0.0.1:8000/api/v1/users/create\_supervisor" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJhZG1pbiIsInJvbGUiOiJBRE1JTiIsInVzZXJfaWQiOjEsImV4cCI6MTc0NTYxODk4MX0.u9phRfBtNTaK6WZLTIjiTtHje9WX0lloeV16h\_4-0Pc" ^

More? -H "Content-Type: application/json" ^

More? -d "{\"username\":\"supervisor123\",\"email\":\"supervisor123@example.com\",\"password\":\"SuperSecure123\",\"first\_name\":\"John\",\"last\_name\":\"Doe\",\"department\_role\":\"USPS\_SUPERVISOR\",\"department\_id\":1,\"is\_active\":true}"

{"message":"Supervisor supervisor123 created successfully","employee\_id":"EMP009"}

1. **Add department as Admin - Namitha**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl.exe -X POST "http://127.0.0.1:8000/api/v1/departments/create\_department" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJhZG1pbiIsInJvbGUiOiJBRE1JTiIsInVzZXJfaWQiOjEsImV4cCI6MTc0NTYyNDQwNn0.F9bVBSdy8iwRJZAT-5ztVIr4rV49t0tz9iBWK0sEmDo" ^

More? -H "Content-Type: application/json" ^

More? -d "{\"name\":\"Finance\",\"type\":\"USPS\",\"description\":\"Handles all finanace activities\"}"

{"message":"Department 'Finance' created successfully!","department\_id":3}

**Metric update is restricted for Supervisor**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl -X POST http://localhost:8000/api/v1/metrics/update-metric ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ1Njk5NjYwfQ.dSVRgu4Qj63jQvK5CRCZ77q5-fF8ffxCuwwlnhFpBaM" ^

More? -H "Content-Type: application/json" ^

More? -d "{\"metric\_id\": 6, \"value\_text\": \"No injuries today\"}"

{"detail":"Only employees can update metrics."}

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>

**Employee can view his own details**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl -X GET http://localhost:8000/api/v1/profile/me ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJwYXRyaWNrIiwicm9sZSI6IkVNUExPWUVFIiwidXNlcl9pZCI6NCwiZXhwIjoxNzQ1NzA5ODQxfQ.b9O2GFf2hhs92uL0FC7TTX22e9tnMn15AZ6pKEqZS8c"

{"email":"patrick@example.com","first\_name":"Patrick","last\_name":"Smith","role":"EMPLOYEE","department\_role":"USPS\_MAIL\_CARRIER","department\_id":1,"employee\_id":"EMP004"}

**Supervisor updating employee metric**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl -X POST http://localhost:8000/api/v1/metric-records/supervisor-update-metric?employee\_id=5 ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ1NzE0Nzg3fQ.CEyTJ8YoU74Q4VOqY9JNTAOSBn5zGEZoo8p0F3Fqldw" ^

More? -H "Content-Type: application/json" ^

More? -d "{\"metric\_id\": 14, \"value\_numeric\": 5}"

{"message":"Employee metric updated successfully by Supervisor."}

**Supervisor try to update other department metric**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem\backend>curl -X POST http://localhost:8000/api/v1/metric-records/supervisor-update-metric?employee\_id=5 ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ1NzE0Nzg3fQ.CEyTJ8YoU74Q4VOqY9JNTAOSBn5zGEZoo8p0F3Fqldw" ^

More? -H "Content-Type: application/json" ^

More? -d "{\"metric\_id\": 21, \"value\_numeric\": 5}"

{"detail":"You are not allowed to edit this metric."}

**Employee updating metric in bulk**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem>curl -X POST http://localhost:8000/api/v1/metric-records/employee-submit-metrics ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJwYXRyaWNrIiwicm9sZSI6IkVNUExPWUVFIiwidXNlcl9pZCI6NCwiZXhwIjoxNzQ1OTUzNjczfQ.1CKxQrwMKAi8sRMlpgkPeFTo1Tlz15eVaxelbtbLyC4" ^

More? -H "Content-Type: application/json" ^

More? -d "{\"metrics\": [{\"metric\_id\": 1, \"value\_numeric\": 50},{\"metric\_id\": 3, \"value\_numeric\": 4},{\"metric\_id\": 4, \"value\_numeric\": 2},{\"metric\_id\": 5, \"value\_numeric\": 15},{\"metric\_id\": 6, \"value\_text\": \"Met with an accident\"},{\"metric\_id\": 9, \"value\_numeric\": 3}]}"

{"message":"All metrics submitted successfully."}

**Supervisor to view the metric of the employee**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem>curl -X GET http://localhost:8000/api/v1/metric-records/employee/4/metrics ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ1OTU0OTAzfQ.ZA8T14TgJ1KTz-8e0qPdsq5U-m0ql0mLtMKliBEPH8I"

[{"id":2,"metric\_type":"performance","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.804919-07:00","metric\_id":1,"user\_id":4,"value\_numeric":50.0,"value\_json":null,"notes":null},{"id":3,"metric\_type":"performance","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.808708-07:00","metric\_id":3,"user\_id":4,"value\_numeric":4.0,"value\_json":null,"notes":null},{"id":4,"metric\_type":"performance","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.810713-07:00","metric\_id":4,"user\_id":4,"value\_numeric":2.0,"value\_json":null,"notes":null},{"id":5,"metric\_type":"wellness","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.812731-07:00","metric\_id":5,"user\_id":4,"value\_numeric":15.0,"value\_json":null,"notes":null},{"id":6,"metric\_type":"wellness","value\_text":"Met with an accident","recorded\_at":"2025-04-29T11:44:32.814736-07:00","metric\_id":6,"user\_id":4,"value\_numeric":null,"value\_json":null,"notes":null},{"id":7,"metric\_type":"wellness","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.816740-07:00","metric\_id":9,"user\_id":4,"value\_numeric":3.0,"value\_json":null,"notes":null}]

**Employee to view his own performance metric**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem>curl -X GET http://localhost:8000/api/v1/metric-records/my-performance-metrics ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJwYXRyaWNrIiwicm9sZSI6IkVNUExPWUVFIiwidXNlcl9pZCI6NCwiZXhwIjoxNzQ1OTU2NTg2fQ.9WHzpeAH1Ysgs45ToDhSIrxjIwYwJMNa13GHfnJCMy0"

[{"id":2,"metric\_type":"performance","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.804919-07:00","metric\_id":1,"user\_id":4,"value\_numeric":50.0,"value\_json":null,"notes":null},{"id":3,"metric\_type":"performance","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.808708-07:00","metric\_id":3,"user\_id":4,"value\_numeric":4.0,"value\_json":null,"notes":null},{"id":4,"metric\_type":"performance","value\_text":null,"recorded\_at":"2025-04-29T11:44:32.810713-07:00","metric\_id":4,"user\_id":4,"value\_numeric":2.0,"value\_json":null,"notes":null}]

**Supervisor to search the employee by employee-id of his own department**

C:\Users\shaba>curl -X GET "http://localhost:8000/api/v1/metric-records/employee/search-by-id/EMP004" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NTU2Nzc1fQ.gUDM7yT-iG-NAEmpIX0Fd2T9UrWbLSWUfdmfUoZZNNM" ^

More? -H "Accept: application/json"

{"employee\_id":"EMP004","full\_name":"Patrick Smith","email":"patrick@example.com","department\_id":1,"performance\_metrics":[{"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":50.0,"value\_text":null,"recorded\_at":"2025-04-29"},{"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-04-29"},{"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-04-29"},{"metric\_name":"Customer Satisfaction Score","metric\_type":"performance","value\_numeric":5.0,"value\_text":null,"recorded\_at":"2025-04-30"},{"metric\_name":"Aggregated Customer Feedback","metric\_type":"performance","value\_numeric":null,"value\_text":"Excellent service","recorded\_at":"2025-04-30"}],"wellness\_metrics":[{"metric\_name":"Distance Covered","metric\_type":"wellness","value\_numeric":15.0,"value\_text":null,"recorded\_at":"2025-04-29"},{"metric\_name":"Injury Report","metric\_type":"wellness","value\_numeric":null,"value\_text":"Met with an accident","recorded\_at":"2025-04-29"},{"metric\_name":"Stress Level","metric\_type":"wellness","value\_numeric":3.0,"value\_text":null,"recorded\_at":"2025-04-29"}]}

**Negative Scenario - When tries to search other department employee**  
C:\Users\shaba>curl -X GET   
"http://localhost:8000/api/v1/metric-records/employee/search-by-id/EMP005" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NTU2Nzc1fQ.gUDM7yT-iG-NAEmpIX0Fd2T9UrWbLSWUfdmfUoZZNNM" ^

More? -H "Accept: application/json"

{"detail":"You can only view employees in your department."}

**Supervisor to list all the employees data of his department**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem>curl -X GET "http://localhost:8000/api/v1/metric-records/department/employee-metrics" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NTYwMzY2fQ.h2ROQqDZMJObY-AUDnyAjRSWYKwyPuDyc1R4IRwZZzQ" ^

More? -H "Accept: application/json"

[{"employee\_id":"EMP004","first\_name":"Patrick","last\_name":"Smith","metrics":[{"metric\_id":15,"metric\_name":"Aggregated Customer Feedback","metric\_type":"performance","value\_numeric":null,"value\_text":"Excellent service","recorded\_at":"2025-04-30","unit":"Text"},{"metric\_id":14,"metric\_name":"Customer Satisfaction Score","metric\_type":"performance","value\_numeric":5.0,"value\_text":null,"recorded\_at":"2025-04-30","unit":"Score"},{"metric\_id":9,"metric\_name":"Stress Level","metric\_type":"wellness","value\_numeric":3.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Score"},{"metric\_id":6,"metric\_name":"Injury Report","metric\_type":"wellness","value\_numeric":null,"value\_text":"Met with an accident","recorded\_at":"2025-04-29","unit":"Text"},{"metric\_id":5,"metric\_name":"Distance Covered","metric\_type":"wellness","value\_numeric":15.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Miles"},{"metric\_id":4,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"},{"metric\_id":3,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"},{"metric\_id":1,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":50.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"}]},{"employee\_id":"EMP007","first\_name":"Robert","last\_name":"Ross","metrics":[{"metric\_id":14,"metric\_name":"Customer Satisfaction Score","metric\_type":"performance","value\_numeric":5.0,"value\_text":null,"recorded\_at":"2025-04-26","unit":"Score"}]}]

* **View all employee performance metrics by date range**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem>curl -X GET "http://localhost:8000/api/v1/metric-records/department/employee-metrics?metric\_type=PERFORMANCE&start\_date=2025-04-26&end\_date=2025-04-29" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NTYwMzY2fQ.h2ROQqDZMJObY-AUDnyAjRSWYKwyPuDyc1R4IRwZZzQ" ^

More? -H "Accept: application/json"

[{"employee\_id":"EMP004","first\_name":"Patrick","last\_name":"Smith","metrics":[{"metric\_id":4,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"},{"metric\_id":3,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"},{"metric\_id":1,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":50.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"}]},{"employee\_id":"EMP007","first\_name":"Robert","last\_name":"Ross","metrics":[{"metric\_id":14,"metric\_name":"Customer Satisfaction Score","metric\_type":"performance","value\_numeric":5.0,"value\_text":null,"recorded\_at":"2025-04-26","unit":"Score"}]}]

* **View All employees - performance + wellness metrics by year**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem>curl -X GET "http://localhost:8000/api/v1/metric-records/department/employee-metrics?year=2025&metric\_type=PERFORMANCE" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NTYwMzY2fQ.h2ROQqDZMJObY-AUDnyAjRSWYKwyPuDyc1R4IRwZZzQ" ^

More? -H "Accept: application/json"

[{"employee\_id":"EMP004","first\_name":"Patrick","last\_name":"Smith","metrics":[{"metric\_id":14,"metric\_name":"Customer Satisfaction Score","metric\_type":"performance","value\_numeric":5.0,"value\_text":null,"recorded\_at":"2025-04-30","unit":"Score"},{"metric\_id":15,"metric\_name":"Aggregated Customer Feedback","metric\_type":"performance","value\_numeric":null,"value\_text":"Excellent service","recorded\_at":"2025-04-30","unit":"Text"},{"metric\_id":4,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"},{"metric\_id":3,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"},{"metric\_id":1,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":50.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Count"}]},{"employee\_id":"EMP007","first\_name":"Robert","last\_name":"Ross","metrics":[{"metric\_id":14,"metric\_name":"Customer Satisfaction Score","metric\_type":"performance","value\_numeric":5.0,"value\_text":null,"recorded\_at":"2025-04-26","unit":"Score"}]}]

* **View all employees Wellness metric by month + year**

D:\SJSU\CMPE\_272\Project\EmployeePerfMonitoringSystem>curl -X GET "http://localhost:8000/api/v1/metric-records/department/employee-metrics?year=2025&month=4&metric\_type=WELLNESS" ^

More? -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NTYwMzY2fQ.h2ROQqDZMJObY-AUDnyAjRSWYKwyPuDyc1R4IRwZZzQ" ^

More? -H "Accept: application/json"

[{"employee\_id":"EMP004","first\_name":"Patrick","last\_name":"Smith","metrics":[{"metric\_id":9,"metric\_name":"Stress Level","metric\_type":"wellness","value\_numeric":3.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Score"},{"metric\_id":6,"metric\_name":"Injury Report","metric\_type":"wellness","value\_numeric":null,"value\_text":"Met with an accident","recorded\_at":"2025-04-29","unit":"Text"},{"metric\_id":5,"metric\_name":"Distance Covered","metric\_type":"wellness","value\_numeric":15.0,"value\_text":null,"recorded\_at":"2025-04-29","unit":"Miles"}]}]

Flow Based Testing

Flow Based Testing

1. **Flow 1 - Employee login , add data , view my metrics and data.**

Employee Login - View Profile , Add perf data , Add wellness data , View perf metrics ,View wellness metrics - logout

* **Login as Employee patrick**

TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-d "username=patrick&password=patrick123" | jq -r .access\_token)

TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-d "username=jane&password=jane123" | jq -r .access\_token)

echo $TOKENVAL

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJwYXRyaWNrIiwicm9sZSI6IkVNUExPWUVFIiwidXNlcl9pZCI6NCwiZXhwIjoxNzQ2MzE1MDYwfQ.bCsmtvELnxrlc2YFH8J3KwQsHmYKntkZ2J77OZ2z3WQ

* **View My Profile**

curl -X GET "http://localhost:8000/api/v1/profile/me" \

-H "accept: application/json" \

-H "Authorization: Bearer $TOKENVAL"

{"email":"patrick@example.com","first\_name":"Patrick","last\_name":"Smith","role":"EMPLOYEE","department\_role":"1","department\_id":1,"department":{"id":1,"name":"US Postal Service","type":"USPS","description":"Handles mail delivery"},"employee\_id":"EMP004"}**%**

curl -X GET "http://localhost:8000/api/v1/profile/me" \

-H "accept: application/json" \

-H "Authorization: Bearer $TOKENVAL"

{"email":"jane@example.com","first\_name":"Jane","last\_name":"Doe","role":"EMPLOYEE","department\_role":"HEALTHCARE\_NURSE","department\_id":2,"department":{"id":2,"name":"Healthcare","type":"HEALTHCARE","description":"Federal health services"},"employee\_id":"EMP005"}**%**

* **View all available metrics - based on this front end must render page as per department, roletype (employee/supervisor) and department role\_type (mail carrier , health admin etc)**

Table relations:

wellness\_db=# SELECT id, username, role\_id from users;

id | username | role\_id

----+----------+---------

1 | admin |

2 | jason | 7

3 | johnny | 6

4 | patrick | 1

5 | Robert | 2

6 | jane | 3

7 | michael | 4

wellness\_db=# SELECT \*from employee\_roles;

role\_id | role\_name | role\_description

---------+---------------------------+--------------------------------------------------------

1 | USPS\_MAIL\_CARRIER | Postal worker responsible for mail and parcel delivery

2 | USPS\_OFFICE\_ADMIN | Administrative staff in USPS office

3 | USPS\_SUPERVISOR | Supervisor in USPS department

4 | HEALTHCARE\_NURSE | Registered Nurse

5 | HEALTHCARE\_ADMIN | Healthcare administrative staff

6 | HEALTHCARE\_SUPERVISOR | Healthcare department supervisor

madhunica@Madhunicas-MacBook-Air ~ % curl -X GET http://localhost:8000/api/v1/metric-records/employee/available-metrics \

-H "Authorization: Bearer $TOKENVAL" \

-H "Accept: application/json"

[{"id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","unit":"Count"},{"id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","unit":"Count"},{"id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","unit":"Count"},{"id":5,"metric\_name":"Redelivery Attempts","metric\_type":"performance","unit":"Count"},{"id":6,"metric\_name":"Distance Covered","metric\_type":"wellness","unit":"Miles"},{"id":7,"metric\_name":"Injury Report","metric\_type":"wellness","unit":"Text"},{"id":8,"metric\_name":"Weather Exposure","metric\_type":"wellness","unit":"Severity"},{"id":9,"metric\_name":"Sick Days","metric\_type":"wellness","unit":"Count"},{"id":10,"metric\_name":"Stress Level","metric\_type":"wellness","unit":"Score"},{"id":11,"metric\_name":"Work-Life Balance","metric\_type":"wellness","unit":"Score"},{"id":12,"metric\_name":"Job Satisfaction","metric\_type":"wellness","unit":"Score"}]**%** madhunica@Madhunicas-MacBook-Air ~ %

madhunica@Madhunicas-MacBook-Air backend % curl -X GET http://localhost:8000/api/v1/metric-records/employee/available-metrics \

-H "Authorization: Bearer $TOKENVAL" \

-H "Accept: application/json"

[{"id":21,"metric\_name":"Patient Satisfaction Score","metric\_type":"performance","unit":"Score"},{"id":22,"metric\_name":"Patient Feedback","metric\_type":"performance","unit":"Score"},{"id":23,"metric\_name":"Vaccinations Administered","metric\_type":"performance","unit":"Count"},{"id":27,"metric\_name":"Night Shift Hours Worked","metric\_type":"wellness","unit":"Hours"},{"id":29,"metric\_name":"Medication Errors","metric\_type":"performance","unit":"Count"},{"id":30,"metric\_name":"Clinical Procedures Completed","metric\_type":"performance","unit":"Count"},{"id":31,"metric\_name":"Breaks Taken On Time","metric\_type":"wellness","unit":"Count"},{"id":32,"metric\_name":"Physical Strain Reports","metric\_type":"wellness","unit":"Count"}]**%** madhunica@Madhunicas-MacBook-Air backend %

* **View applicable wellness metrics - based on role type in my department(useful for frontend)**

madhunica@Madhunicas-MacBook-Air ~ % curl -X GET http://localhost:8000/api/v1/metric-records/employee/wellness-metrics \

-H "Authorization: Bearer $TOKENVAL" \

-H "Accept: application/json"

[{"id":6,"metric\_name":"Distance Covered","metric\_type":"wellness","unit":"Miles"},{"id":7,"metric\_name":"Injury Report","metric\_type":"wellness","unit":"Text"},{"id":8,"metric\_name":"Weather Exposure","metric\_type":"wellness","unit":"Severity"},{"id":9,"metric\_name":"Sick Days","metric\_type":"wellness","unit":"Count"},{"id":10,"metric\_name":"Stress Level","metric\_type":"wellness","unit":"Score"},{"id":11,"metric\_name":"Work-Life Balance","metric\_type":"wellness","unit":"Score"},{"id":12,"metric\_name":"Job Satisfaction","metric\_type":"wellness","unit":"Score"}]**%**

* **View applicable performance metrics - based on role type in my department(useful for frontend)**

madhunica@Madhunicas-MacBook-Air ~ % curl -X GET http://localhost:8000/api/v1/metric-records/employee/performance-metrics \

-H "Authorization: Bearer $TOKENVAL" \

-H "Accept: application/json"

[{"id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","unit":"Count"},{"id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","unit":"Count"},{"id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","unit":"Count"},{"id":5,"metric\_name":"Redelivery Attempts","metric\_type":"performance","unit":"Count"}]**%** madhunica@Madhunicas-MacBook-Air ~ %

* **Add performance & wellness data - maintain granularity based on date - USPS Mail carrier**

madhunica@Madhunicas-MacBook-Air ~ % curl -X POST http://localhost:8000/api/v1/metric-records/employee-submit-metrics \

-H "Authorization: Bearer $TOKENVAL" \

-H "Content-Type: application/json" \

-d '{

"date": "2025-05-05",

"metrics": [

{ "metric\_id": 2, "value\_numeric": 100 },

{ "metric\_id": 3, "value\_numeric": 4 },

{ "metric\_id": 4, "value\_numeric": 1 },

{ "metric\_id": 5, "value\_numeric": 2 },

{ "metric\_id": 6, "value\_numeric": 15.2 },

{ "metric\_id": 7, "value\_text": "None" },

{ "metric\_id": 8, "value\_text": "High" },

{ "metric\_id": 9, "value\_numeric": 0 },

{ "metric\_id": 10, "value\_numeric": 6 },

{ "metric\_id": 11, "value\_numeric": 7 },

{ "metric\_id": 12, "value\_numeric": 8 }

]

}'

{"message":"Metrics submitted for 2025-05-05"}**%**

wellness\_db=# SELECT \*from metric\_records;

id | user\_id | metric\_id | metric\_type | value\_numeric | value\_json | value\_text | recorded\_at | notes

----+---------+-----------+-------------+---------------+------------+------------+------------------------+-------

1 | 4 | 2 | PERFORMANCE | 100 | | | 2025-05-05 00:00:00-07 |

2 | 4 | 3 | PERFORMANCE | 4 | | | 2025-05-05 00:00:00-07 |

3 | 4 | 4 | PERFORMANCE | 1 | | | 2025-05-05 00:00:00-07 |

4 | 4 | 5 | PERFORMANCE | 2 | | | 2025-05-05 00:00:00-07 |

5 | 4 | 6 | WELLNESS | 15.2 | | | 2025-05-05 00:00:00-07 |

6 | 4 | 7 | WELLNESS | | | None | 2025-05-05 00:00:00-07 |

7 | 4 | 8 | WELLNESS | | | High | 2025-05-05 00:00:00-07 |

8 | 4 | 9 | WELLNESS | 0 | | | 2025-05-05 00:00:00-07 |

9 | 4 | 10 | WELLNESS | 6 | | | 2025-05-05 00:00:00-07 |

10 | 4 | 11 | WELLNESS | 7 | | | 2025-05-05 00:00:00-07 |

11 | 4 | 12 | WELLNESS | 8 | | | 2025-05-05 00:00:00-07 |

* **Add performance & wellness data - maintain granularity based on date - USPS Office Admin**

madhunica@Madhunicas-MacBook-Air ~ % curl -X POST http://localhost:8000/api/v1/metric-records/employee-submit-metrics \

-H "Authorization: Bearer $TOKENVAL" \

-H "Content-Type: application/json" \

-d '{

"date": "2025-05-05",

"metrics": [

{ "metric\_id": 9, "value\_numeric": 1 },

{ "metric\_id": 10, "value\_numeric": 6 },

{ "metric\_id": 11, "value\_numeric": 7 },

{ "metric\_id": 12, "value\_numeric": 8 },

{ "metric\_id": 13, "value\_numeric": 20 },

{ "metric\_id": 14, "value\_numeric": 18 },

{ "metric\_id": 15, "value\_numeric": 8 },

{ "metric\_id": 16, "value\_text": "Great service overall." },

{ "metric\_id": 17, "value\_numeric": 25 },

{ "metric\_id": 18, "value\_numeric": 4 }

]

}'

{"message":"Metrics submitted for 2025-05-05"}% madhunica@Madhunicas-MacBook-Air ~ %

12 | 5 | 9 | WELLNESS | 1 | | | 2025-05-05 00:00:00-07 |

13 | 5 | 10 | WELLNESS | 6 | | | 2025-05-05 00:00:00-07 |

14 | 5 | 11 | WELLNESS | 7 | | | 2025-05-05 00:00:00-07 |

15 | 5 | 12 | WELLNESS | 8 | | | 2025-05-05 00:00:00-07 |

16 | 5 | 13 | PERFORMANCE | 20 | | | 2025-05-05 00:00:00-07 |

17 | 5 | 14 | PERFORMANCE | 18 | | | 2025-05-05 00:00:00-07 |

18 | 5 | 15 | PERFORMANCE | 8 | | | 2025-05-05 00:00:00-07 |

19 | 5 | 16 | PERFORMANCE | | | Great service overall. | 2025-05-05 00:00:00-07 |

20 | 5 | 17 | PERFORMANCE | 25 | | | 2025-05-05 00:00:00-07 |

21 | 5 | 18 | WELLNESS | 4 | | | 2025-05-05 00:00:00-07 |

* **ViewMy wellness metrics by date range**

madhunica@Madhunicas-MacBook-Air app % curl -X GET "http://localhost:8000/api/v1/metrics/employee/my-metrics?metric\_type=WELLNESS&start\_date=2025-05-01&end\_date=2025-05-05" \

-H "accept: application/json" \

-H "Authorization: Bearer $TOKENVAL"

[{"id":5,"metric\_id":6,"metric\_name":"Distance Covered","metric\_type":"wellness","value\_numeric":15.2,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Miles"},{"id":6,"metric\_id":7,"metric\_name":"Injury Report","metric\_type":"wellness","value\_numeric":null,"value\_text":"None","recorded\_at":"2025-05-05","unit":"Text"},{"id":7,"metric\_id":8,"metric\_name":"Weather Exposure","metric\_type":"wellness","value\_numeric":null,"value\_text":"High","recorded\_at":"2025-05-05","unit":"Severity"},{"id":8,"metric\_id":9,"metric\_name":"Sick Days","metric\_type":"wellness","value\_numeric":0.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":9,"metric\_id":10,"metric\_name":"Stress Level","metric\_type":"wellness","value\_numeric":6.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Score"},{"id":10,"metric\_id":11,"metric\_name":"Work-Life Balance","metric\_type":"wellness","value\_numeric":7.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Score"},{"id":11,"metric\_id":12,"metric\_name":"Job Satisfaction","metric\_type":"wellness","value\_numeric":8.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Score"},{"id":30,"metric\_id":6,"metric\_name":"Distance Covered","metric\_type":"wellness","value\_numeric":null,"value\_text":"13","recorded\_at":"2025-05-02","unit":"Miles"},{"id":31,"metric\_id":7,"metric\_name":"Injury Report","metric\_type":"wellness","value\_numeric":null,"value\_text":"0","recorded\_at":"2025-05-02","unit":"Text"},{"id":32,"metric\_id":8,"metric\_name":"Weather Exposure","metric\_type":"wellness","value\_numeric":null,"value\_text":"1","recorded\_at":"2025-05-02","unit":"Severity"},{"id":33,"metric\_id":9,"metric\_name":"Sick Days","metric\_type":"wellness","value\_numeric":null,"value\_text":"1","recorded\_at":"2025-05-02","unit":"Count"},{"id":34,"metric\_id":10,"metric\_name":"Stress Level","metric\_type":"wellness","value\_numeric":null,"value\_text":"Low","recorded\_at":"2025-05-02","unit":"Score"},{"id":35,"metric\_id":11,"metric\_name":"Work-Life Balance","metric\_type":"wellness","value\_numeric":null,"value\_text":"Happy","recorded\_at":"2025-05-02","unit":"Score"},{"id":36,"metric\_id":12,"metric\_name":"Job Satisfaction","metric\_type":"wellness","value\_numeric":null,"value\_text":"Very Satisfied","recorded\_at":"2025-05-02","unit":"Score"}]**%** madhunica@Madhunicas-MacBook-Air app %

* **ViewMy performance metrics by date range**

madhunica@Madhunicas-MacBook-Air app % curl -X GET "http://localhost:8000/api/v1/metrics/employee/my-metrics?metric\_type=PERFORMANCE&start\_date=2025-05-01&end\_date=2025-05-05" \

-H "accept: application/json" \

-H "Authorization: Bearer $TOKENVAL"

[{"id":1,"metric\_id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":100.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":2,"metric\_id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":3,"metric\_id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":1.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":4,"metric\_id":5,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":26,"metric\_id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":1.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"},{"id":27,"metric\_id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"},{"id":28,"metric\_id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"},{"id":29,"metric\_id":5,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":6.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"}]**%** madhunica@Madhunicas-MacBook-Air app %

* **View All - performance + wellness metrics by month & year**

madhunica@Madhunicas-MacBook-Air app % curl -X GET "http://localhost:8000/api/v1/metrics/employee/my-metrics?month=5&year=2025" \

-H "accept: application/json" \

-H "Authorization: Bearer $TOKENVAL"

[{"id":22,"metric\_id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":13.0,"value\_text":null,"recorded\_at":"2025-05-06","unit":"Count"},{"id":23,"metric\_id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-05-06","unit":"Count"},{"id":24,"metric\_id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":3.0,"value\_text":null,"recorded\_at":"2025-05-06","unit":"Count"},{"id":25,"metric\_id":5,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":1.0,"value\_text":null,"recorded\_at":"2025-05-06","unit":"Count"},{"id":5,"metric\_id":6,"metric\_name":"Distance Covered","metric\_type":"wellness","value\_numeric":15.2,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Miles"},{"id":6,"metric\_id":7,"metric\_name":"Injury Report","metric\_type":"wellness","value\_numeric":null,"value\_text":"None","recorded\_at":"2025-05-05","unit":"Text"},{"id":7,"metric\_id":8,"metric\_name":"Weather Exposure","metric\_type":"wellness","value\_numeric":null,"value\_text":"High","recorded\_at":"2025-05-05","unit":"Severity"},{"id":8,"metric\_id":9,"metric\_name":"Sick Days","metric\_type":"wellness","value\_numeric":0.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":9,"metric\_id":10,"metric\_name":"Stress Level","metric\_type":"wellness","value\_numeric":6.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Score"},{"id":10,"metric\_id":11,"metric\_name":"Work-Life Balance","metric\_type":"wellness","value\_numeric":7.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Score"},{"id":11,"metric\_id":12,"metric\_name":"Job Satisfaction","metric\_type":"wellness","value\_numeric":8.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Score"},{"id":1,"metric\_id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":100.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":2,"metric\_id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":3,"metric\_id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":1.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":4,"metric\_id":5,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-05-05","unit":"Count"},{"id":36,"metric\_id":12,"metric\_name":"Job Satisfaction","metric\_type":"wellness","value\_numeric":null,"value\_text":"Very Satisfied","recorded\_at":"2025-05-02","unit":"Score"},{"id":32,"metric\_id":8,"metric\_name":"Weather Exposure","metric\_type":"wellness","value\_numeric":null,"value\_text":"1","recorded\_at":"2025-05-02","unit":"Severity"},{"id":33,"metric\_id":9,"metric\_name":"Sick Days","metric\_type":"wellness","value\_numeric":null,"value\_text":"1","recorded\_at":"2025-05-02","unit":"Count"},{"id":34,"metric\_id":10,"metric\_name":"Stress Level","metric\_type":"wellness","value\_numeric":null,"value\_text":"Low","recorded\_at":"2025-05-02","unit":"Score"},{"id":35,"metric\_id":11,"metric\_name":"Work-Life Balance","metric\_type":"wellness","value\_numeric":null,"value\_text":"Happy","recorded\_at":"2025-05-02","unit":"Score"},{"id":30,"metric\_id":6,"metric\_name":"Distance Covered","metric\_type":"wellness","value\_numeric":null,"value\_text":"13","recorded\_at":"2025-05-02","unit":"Miles"},{"id":31,"metric\_id":7,"metric\_name":"Injury Report","metric\_type":"wellness","value\_numeric":null,"value\_text":"0","recorded\_at":"2025-05-02","unit":"Text"},{"id":27,"metric\_id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"},{"id":28,"metric\_id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":4.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"},{"id":29,"metric\_id":5,"metric\_name":"Redelivery Attempts","metric\_type":"performance","value\_numeric":6.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"},{"id":26,"metric\_id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":1.0,"value\_text":null,"recorded\_at":"2025-05-01","unit":"Count"}]**%** madhunica@Madhunicas-MacBook-Air app %

* **View All - performance + wellness metrics by year**

madhunica@Madhunicas-MacBook-Air app % curl -X GET "http://localhost:8000/api/v1/metrics/employee/my-metrics?year=2025" \

-H "accept: application/json" \

-H "Authorization: Bearer $TOKENVAL"

[{"id":22,"metric\_id":2,"metric\_name":"Parcels Delivered On Time","metric\_type":"performance","value\_numeric":13.0,"value\_text":null,"recorded\_at":"2025-05-06","unit":"Count"},{"id":23,"metric\_id":3,"metric\_name":"Parcels Delivered Late","metric\_type":"performance","value\_numeric":2.0,"value\_text":null,"recorded\_at":"2025-05-06","unit":"Count"},{"id":24,"metric\_id":4,"metric\_name":"Parcels Undelivered","metric\_type":"performance","value\_numeric":3.0,"value\_text":null,"recorded\_at":"2025-05-06","unit":"Count"},

**Flow 2 -> Supervisor + Employee data**

**Login as supervisor,**

TOKENVAL=$(curl -X POST "http://localhost:8000/api/v1/auth/login" \

-H "accept: application/json" \

-H "Content-Type: application/x-www-form-urlencoded" \

-d "username=jason&password=jason123" | jq -r .access\_token)

**Add an employee to my department,**

madhunica@Madhunicas-MacBook-Air ~ % curl -X POST http://localhost:8000/api/v1/users/create\_employee \

-H "Content-Type: application/json" \

-H "Authorization: Bearer $TOKENVAL" \

-d '{

"username": "tom",

"email": "tom@example.com",

"password": "Tom!123",

"first\_name": "Tom",

"last\_name": "Cook",

"role": "EMPLOYEE",

"department\_role": 1,

"department\_id": 1,

"role\_id": 1

}'

{"message":"Employee tom created successfully!","employee\_id":"EMP009"}**%** madhunica@Madhunicas-MacBook-Air ~ %

**Update performance metric of an employee - search by employee id ,(Refer employee dashboard) - backend - TBD**

**, frontend - TBD**

**Update wellness metric of an employee - - search by employee id(Refer employee dashboard)**

**backend - TBD , frontend - TBD**

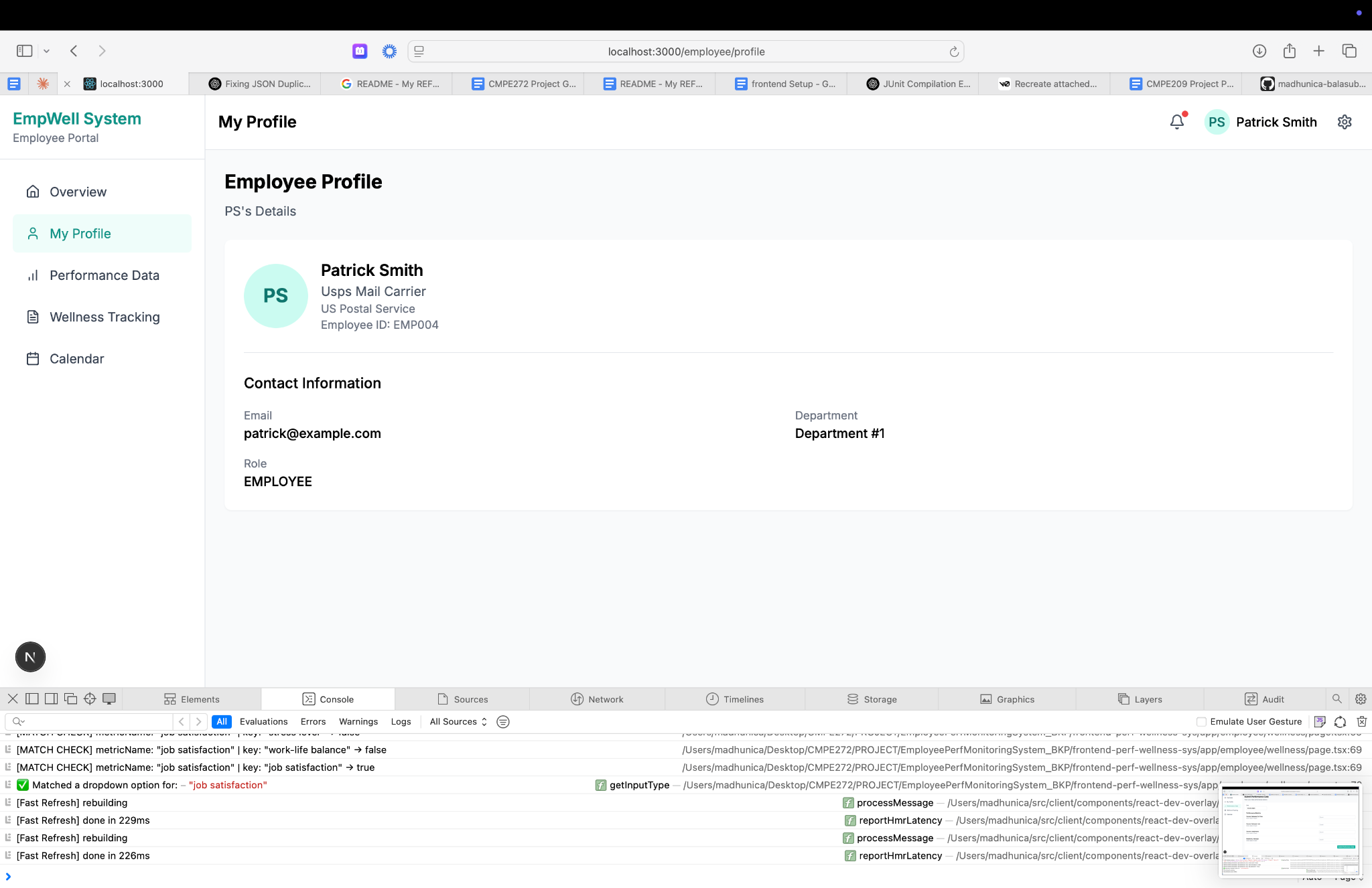
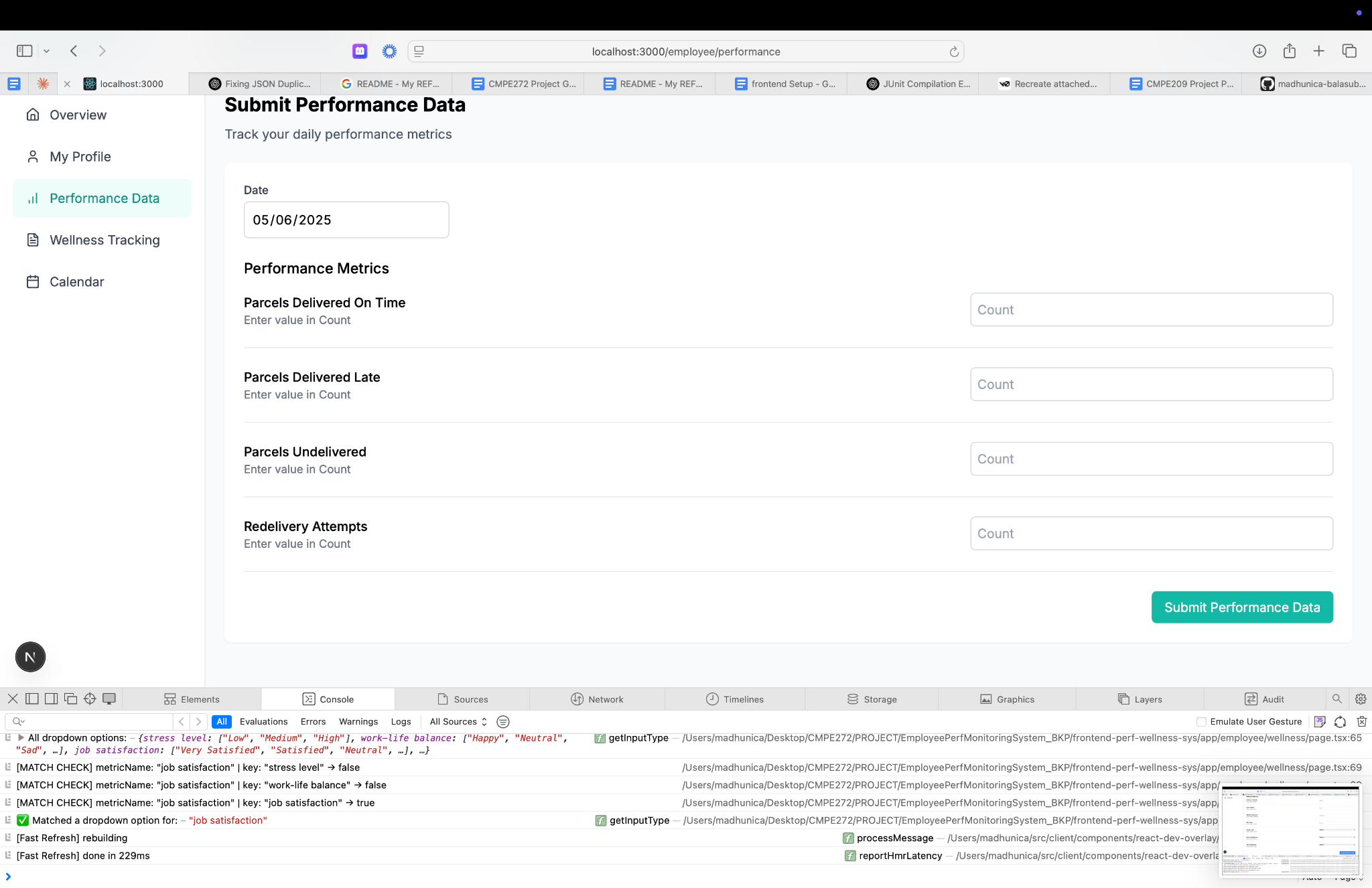
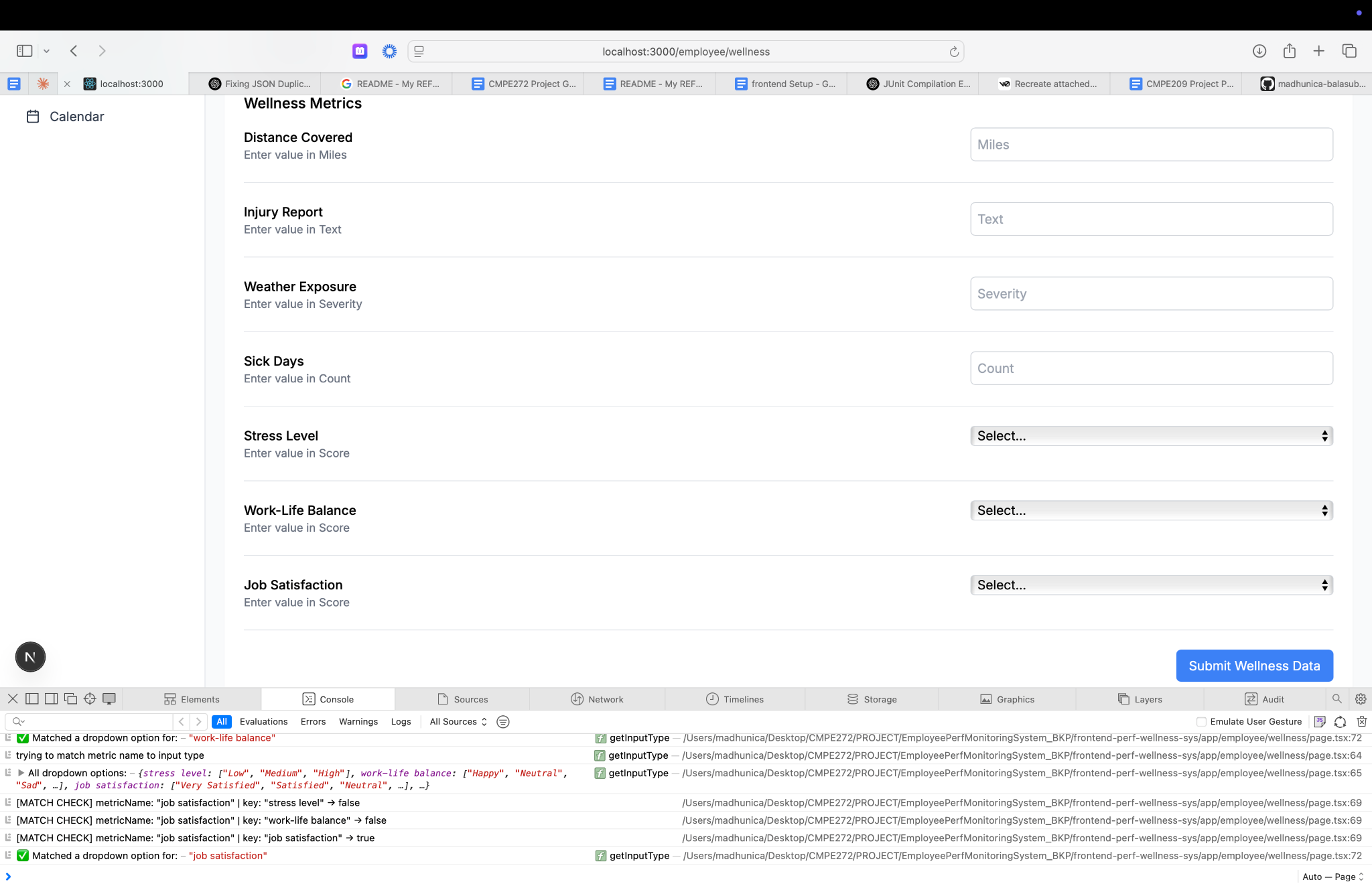
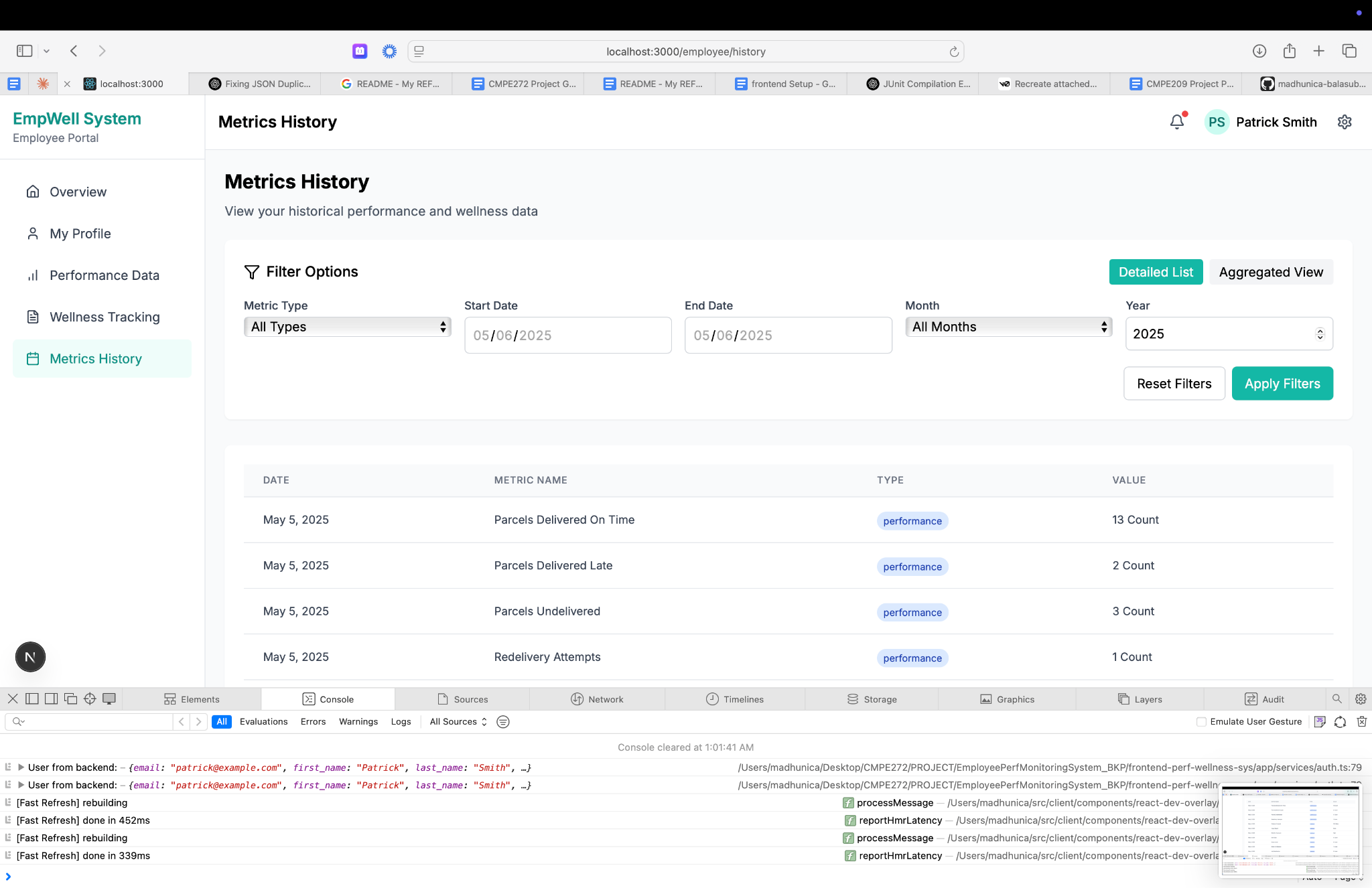
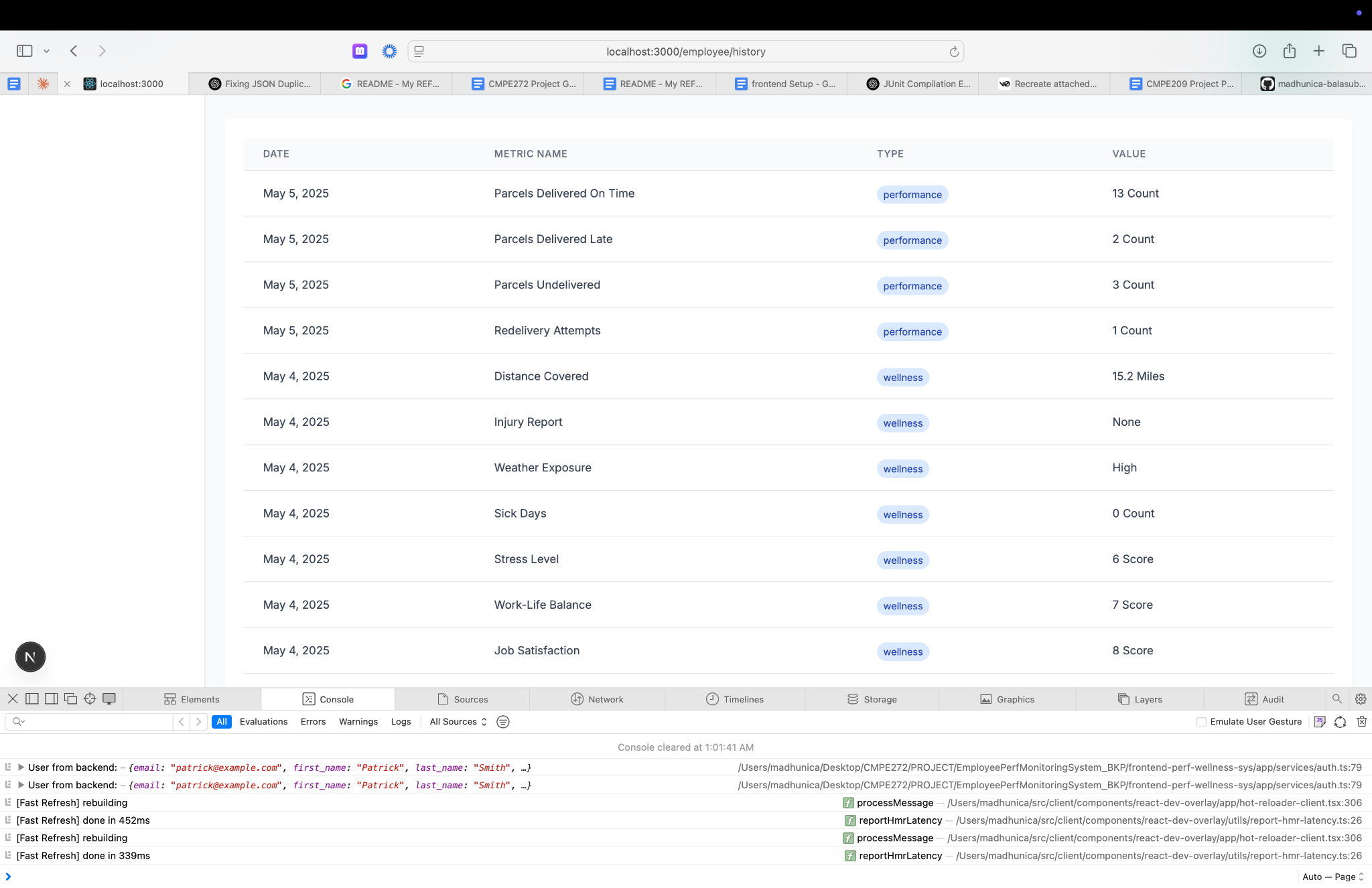
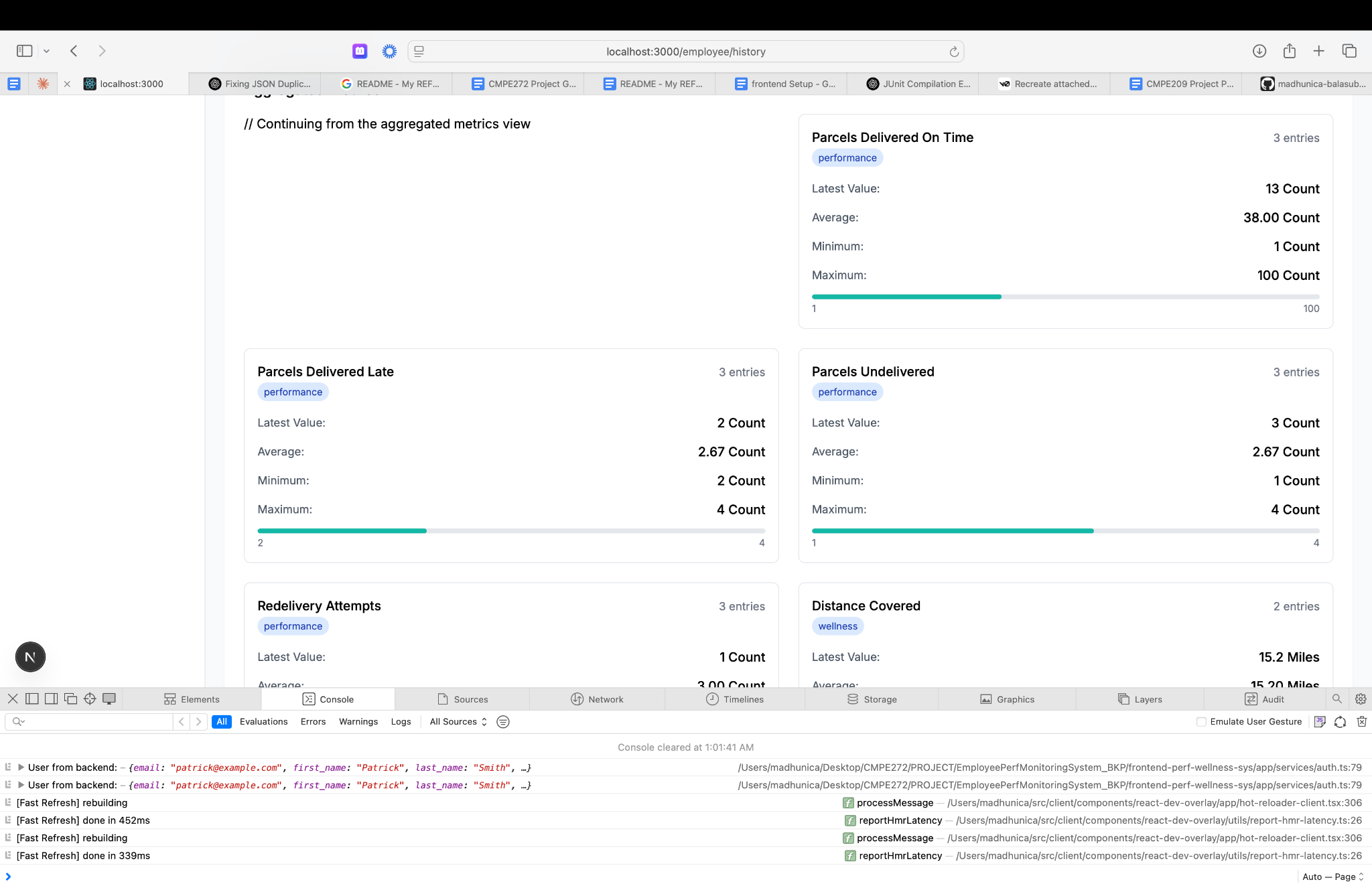
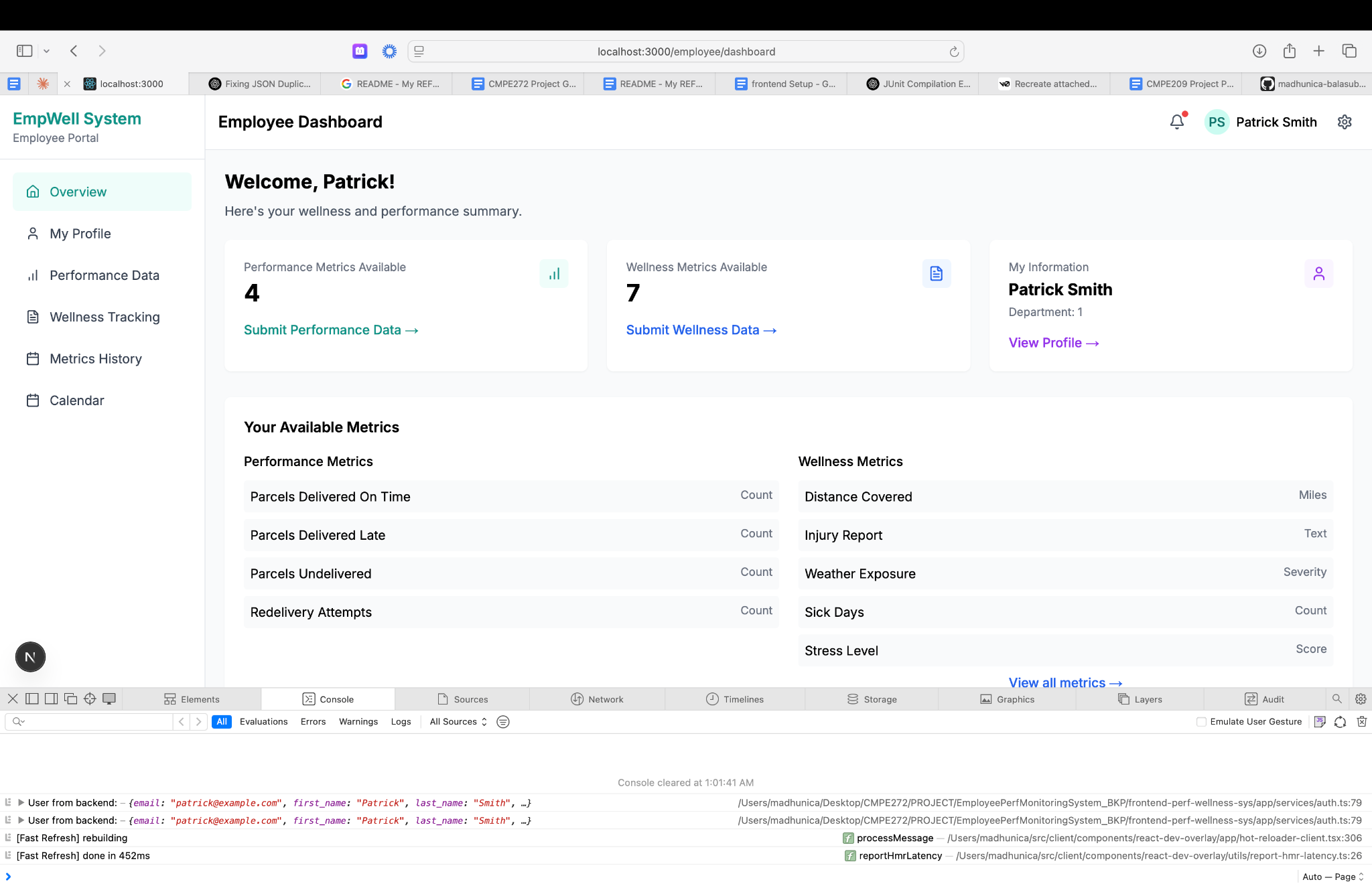
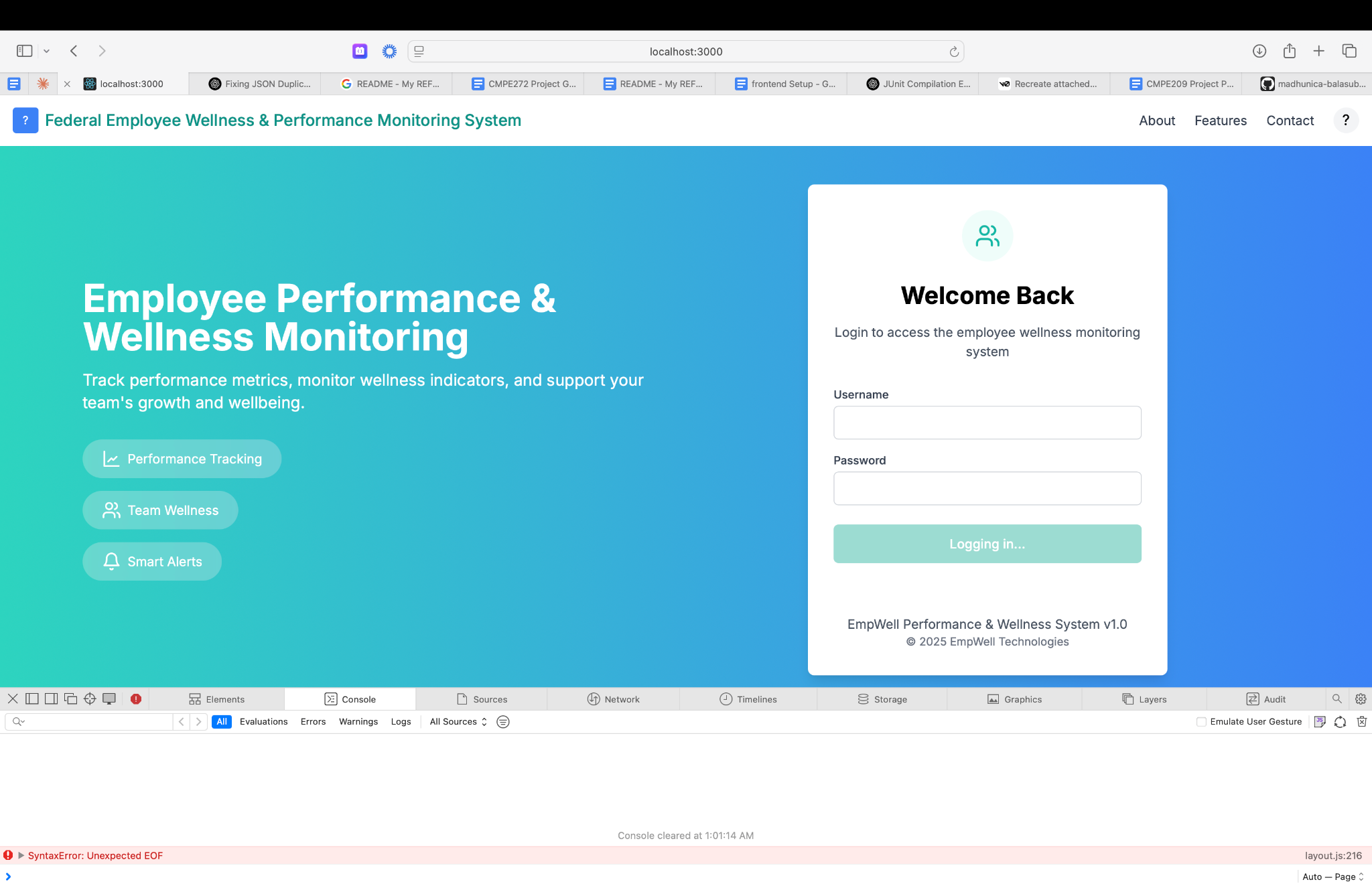
**View all metrics of my department employees (Refer employee dashboard)**

**backend - TBD , , frontend - TBD**

**View all aggregated metrics of my department employees by date, month or year (Refer employee dashboard)**

**backend - TBD , , frontend - TBD**

**Web Page screenshots**



Final Deliverables Status

Final Deliverables

1. Design, Backend, DB design - Madhunica
2. Design, BackEnd - Namita
3. IEEE Paper format - Vignesh
4. FrontEnd & Integration - Madhunica
5. Static Analysis - Namitha
6. Code Coverage - Not Done
7. Professional Video - Madhunica

API Endpoints

# **Employee View Personal Profile**

Method: GET

Full Endpoint: http://localhost:8000/api/v1/profile/me

Payload Required: No

Example Response:

{  
 "email": "patrick@example.com",  
 "first\_name": "Patrick",  
 "last\_name": "Smith",  
 "role": "EMPLOYEE",  
 "department\_role": "USPS\_MAIL\_CARRIER",  
 "department\_id": 1,  
 "employee\_id": "EMP004"  
 }

# **Employee Submit Metrics (Bulk)**

Method: POST

Full Endpoint: http://localhost:8000/api/v1/metric-records/employee-submit-metrics

Payload Required: Yes

Example Request Payload:

{  
 "metrics": [  
 { "metric\_id": 1, "value\_numeric": 50 },  
 { "metric\_id": 3, "value\_numeric": 4 },  
 { "metric\_id": 4, "value\_numeric": 2 },  
 { "metric\_id": 5, "value\_numeric": 15 },  
 { "metric\_id": 6, "value\_text": "Met with an accident" },  
 { "metric\_id": 9, "value\_numeric": 3 }  
 ]  
 }

Example Response:

{  
 "message": "All metrics submitted successfully."  
 }

# **Employee View Own Performance Metrics**

Method: GET

Full Endpoint: http://localhost:8000/api/v1/metric-records/my-performance-metrics

Payload Required: No

# 

# **Employee View Own Wellness Metrics**

Method: GET

Full Endpoint: http://localhost:8000/api/v1/metric-records/my-wellness-metrics

Payload Required: No

# **Supervisor Update Specific Employee's Metric**

Method: POST

Full Endpoint: http://localhost:8000/api/v1/metric-records/supervisor-update-metric?employee\_id={employee\_id}

Payload Required: Yes

Example Request Payload:

{  
 "metrics": [  
 { "metric\_id": 14, "value\_numeric": 5 },  
 { "metric\_id": 15, "value\_text": "Excellent Service" }  
 ]  
 }

Example Response:

{  
 "message": "Employee metric updated successfully by Supervisor."  
 }

# **Supervisor View Employee's Metrics**

Method: GET

Full Endpoint: http://localhost:8000/api/v1/metric-records/employee/{employee\_id}/metrics

Payload Required: No

# **Supervisor Tries to Update Restricted Metric (Error Case)**

Method: POST

Full Endpoint: http://localhost:8000/api/v1/metric-records/supervisor-update-metric?employee\_id={employee\_id}

Payload Required: Yes

Example Request Payload:

{  
 "metric\_id": 21,  
 "value\_numeric": 5  
 }

Example Response:

{  
 "detail": "You are not allowed to edit this metric."  
 }

# **Login - Admin/User/Supervisor**

Method: POST

Full Endpoint: http://127.0.0.1:8000/api/v1/auth/login

Payload Required: Yes

Example Request Payload:

username=admin&password=adminpassword123

Example Response:

{  
 "access\_token": "eyJhbGciOi...",  
 "token\_type": "bearer"  
 }

# **Admin Create Supervisor**

Method: POST

Full Endpoint: http://127.0.0.1:8000/api/v1/users/create\_supervisor

Payload Required: Yes

Example Request Payload:

{  
 "username": "supervisor123",  
 "email": "supervisor123@example.com",  
 "password": "SuperSecurePassword!",  
 "first\_name": "John",  
 "last\_name": "Doe",  
 "department\_role": "USPS\_SUPERVISOR",  
 "department\_id": 1,  
 "is\_active": true  
 }

Example Response:

{  
 "message": "Supervisor supervisor123 created successfully",  
 "employee\_id": "EMP009"  
 }

# **Admin Delete Supervisor**

Method: DELETE

Full Endpoint: http://127.0.0.1:8000/api/v1/users/delete\_supervisor/{supervisor\_id}

Payload Required: No

Example Response:

{  
 "message": "Supervisor johnny deleted successfully."  
 }

# **Admin Create Department**

Method: POST

Full Endpoint: http://127.0.0.1:8000/api/v1/departments/create\_department

Payload Required: Yes

Example Request Payload:

{  
 "name": "Transport",  
 "type": "TRANSPORT",  
 "description": "Handles all transport related activities"  
 }

Example Response:

{  
 "message": "Department 'Transport' created successfully!",  
 "department\_id": 3  
 }

# **Supervisor Create Employee**

Method: POST

Full Endpoint: http://127.0.0.1:8000/api/v1/users/create\_employee

Payload Required: Yes

Example Request Payload:

{  
 "username": "employee001",  
 "email": "employee001@example.com",  
 "password": "StrongPassword123",  
 "first\_name": "John",  
 "last\_name": "Doe",  
 "department\_role": "USPS\_MAIL\_CARRIER",  
 "department\_id": 1,  
 "is\_active": true  
 }

Example Response:

{  
 "message": "Employee Sheldon created successfully!",  
 "employee\_id": "EMP009"  
 }

# **Supervisor Delete Employee**

Method: DELETE

Full Endpoint: http://127.0.0.1:8000/api/v1/users/delete\_employee/{employee\_id}

Payload Required: No

Example Response:

{  
 "message": "Employee Robert deleted successfully."  
 }

**Supervisor to search the employee by employee-Id of his department**

Method : GET

Final endpoint : http://localhost:8000/api/v1/metric-records/employee/search-by-id/EMP005

**Supervisor list the employees of his department with metrics(Date range filters)**

Method : GET

Full Endpoint : http://localhost:8000/api/v1/metric-records/department/employee-metrics

Filters can be added - http://localhost:8000/api/v1/metric-records/department/employee-metrics?year=2025&metric\_type=PERFORMANCE

http://localhost:8000/api/v1/metric-records/department/employee-metrics?metric\_type=PERFORMANCE&start\_date=2025-04-26&end\_date=2025-04-29

Comprehensive Documentation

**Employee Performance Monitoring System**

## **1. System Architecture Overview**

The Employee Wellness & Performance Monitoring System (EWPMS) is built using a modern web architecture with:

* **Backend**: FastAPI (Python-based web framework)
* **Database**: SQL database (via SQLAlchemy ORM)
* **Authentication**: JWT-based authentication

The system follows a typical REST API architecture with clearly defined routes, models, and services.

## **2. Database Models**

### **Core Models**

#### **BaseModel**

* Foundation for all database models
* Provides common fields: id, created\_at, updated\_at
* Implements proper timestamps for data auditing

#### **User**

* Represents system users (employees, supervisors, admins)
* Contains authentication information
* Includes profile details and role information
* Links to department and role-specific information

#### **Department**

* Represents organizational departments (USPS, Healthcare, etc.)
* Includes department type and description
* Contains relationships to users and metrics

#### **Metric-Related Models**

* **MetricDefinition**: Defines different metrics that can be tracked
* **MetricRecord**: Stores actual metric data entries for users
* **EmployeeRole**: Defines specific roles within the system
* **MetricDefinitionRole**: Maps which metrics apply to which roles

## **3. Role-Based Access Control**

The system implements a hierarchical role-based access control system:

### **User Roles**

* **Employee**: Basic access to view and submit their own metrics
* **Supervisor**: Department-level access to manage employees and their metrics
* **Admin**: System-wide access to manage all aspects of the system

### **Department-Specific Roles**

* Different departments have specialized roles (e.g., HEALTHCARE\_NURSE, USPS\_MAIL\_CARRIER)
* Metrics can be tailored to specific department roles

## **4. Metric Types and Tracking**

The system tracks two primary categories of metrics:

### **Performance Metrics**

* Work-related measurements
* Productivity indicators
* Quality assessments
* Department-specific performance indicators

### **Wellness Metrics**

* Health and well-being indicators
* Work-life balance measurements
* Stress and satisfaction levels

Each metric can be:

* Numeric (quantitative values)
* Text-based (qualitative assessments)
* JSON-structured (complex data)

## **5. API Endpoints**

### **Authentication**

* /api/v1/auth/login: User authentication endpoint

### **User Management**

* Create/update/delete employees and supervisors
* View user profiles
* Manage user roles and permissions

### **Department Management**

* Create and manage departments
* View department information
* Assign users to departments

### **Metric Management**

* Define and manage metric definitions
* View available metrics
* Submit and update metric values
* View metric records

## **6. Business Rules**

### **Metric Submission Rules**

* Employees can submit their own metrics but cannot modify certain restricted metrics
* Supervisors can update specific metrics for employees in their department
* Different departments have different rules about which metrics can be modified by whom

### **User Management Rules**

* Admins can create/update/delete supervisors
* Supervisors can only manage employees in their own department
* Employees have limited access to view only their own metrics

## **7. Key Implementation Details**

### **Authentication Flow**

1. User submits credentials
2. System verifies credentials
3. JWT token is generated with user role and permissions
4. Token is used for subsequent API calls

### **Metric Submission Flow**

1. User retrieves available metrics for their department/role
2. User submits metric values
3. System validates submission against business rules
4. Data is stored in metric records

## **8. Git Workflow Notes**

The repository shows evidence of collaborative development:

* Merge conflicts are visible in some files
* Different departmental types are being added by different developers
* Role definitions are evolving over time

## **9. Security Considerations**

* Password hashing for secure storage
* JWT for secure authentication
* Role-based permissions for proper access control
* Input validation on all endpoints

## **10. System Extension Points**

The current architecture allows for:

* Adding new departments and department types
* Creating new metric definitions
* Defining new user roles
* Implementing new business rules for metric submission
* Building dashboards for metric visualization

## **11. Testing Strategies**

For a system like this, consider testing:

* Unit tests for individual components
* Integration tests for API endpoints
* Authentication and authorization testing
* Business rule validation
* Performance testing for metric calculations

## **Architecture Highlights for Presentation**

### **Backend Architecture Highlights**

1. **Clean Layer Separation**
   * **Controller Layer**: FastAPI routes that handle HTTP requests
   * **Service Layer**: Business logic separated from routes
   * **Data Access Layer**: Database models and repositories
   * **Domain Layer**: Core business entities and rules
2. **API-First Design**
   * Endpoint-driven development with Pydantic models
   * Strong type validation using Pydantic schemas
   * Self-documenting API with OpenAPI/Swagger integration
3. **Security Implementation**
   * JWT-based authentication with role-based access control
   * Fine-grained permission model (Employee vs. Supervisor)
   * Secure password handling with proper hashing
4. **Database Design**
   * Entity-relationship model with clear domain boundaries
   * Efficient query patterns for metrics aggregation
   * Proper indexing strategy for performance optimization
5. **Extensibility Features**
   * Metric definition system that can be expanded
   * Department-based customization of metrics
   * Role-metric mapping for personalized experiences

### **Frontend Architecture Highlights**

1. **Component-Based Architecture**
   * Reusable UI components for consistent design
   * Separation of concerns with specialized components
   * Shared layouts with composable elements
2. **Context-Based State Management**
   * Authentication context for user session management
   * Centralized state for user permissions and roles
   * Optimized re-rendering with context selectors
3. **Service Layer Pattern**
   * Abstracted API calls in dedicated service modules
   * Consistent error handling across the application
   * Typed interfaces for API responses
4. **Route Protection Strategy**
   * Role-based access control at route level
   * Centralized protection with the ProtectedRoute component
   * Redirect logic for unauthorized access attempts
5. **Responsive Design Implementation**
   * Mobile-first approach with Tailwind CSS
   * Adaptive layouts for different device sizes
   * Accessibility considerations built-in

*# Install documentation generator* pip install fastapi-codegen *# Generate client code and documentation* fastapi-codegen --input openapi.json --output docs/

npm install -g redoc-cli

npm warn deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.

added 322 packages in 4s

32 packages are looking for funding

run `npm fund` for details

madhunica@Madhunicas-MacBook-Air EmployeePerfMonitoringSystem % redoc-cli bundle http://localhost:8000/openapi.json -o api\_docs.ht

ml

Cloud deployment testing

madhunica@Madhunicas-MacBook-Air backend % curl -X POST "http://54.219.23.135:8000/api/v1/auth/login" \

-H "accept: application/json" \

-H "Content-Type: application/x-www-form-urlencoded" \

-d "username=jane&password=jane123"

{"access\_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYW5lIiwicm9sZSI6IkVNUExPWUVFIiwidXNlcl9pZCI6NiwiZXhwIjoxNzQ2NTY2ODkwfQ.cmQWLEl2ZCUH40jNMyrf3Vs3f3gT1L2PPl20kLOchd8","token\_type":"bearer"}**%** madhunica@Madhunicas-MacBook-Air backend %

D:\>curl -X POST "http://13.57.208.203:8000/api/v1/auth/login" ^

More? -H "Content-Type: application/x-www-form-urlencoded" ^

More? -d "username=jason&password=jason123"

{"access\_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NDE5NTg3fQ.RH530Usd8ooq3xdWBYbj0ewKtjRDanQYOYT9GnFTX4o","token\_type":"bearer"}

D:\>

D:\>curl -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqYXNvbiIsInJvbGUiOiJTVVBFUlZJU09SIiwidXNlcl9pZCI6MiwiZXhwIjoxNzQ2NDE5NTg3fQ.RH530Usd8ooq3xdWBYbj0ewKtjRDanQYOYT9GnFTX4o" http://13.57.208.203:8000/api/v1/profile/me

{"email":"jason@example.com","first\_name":"Jason","last\_name":"Smith","role":"SUPERVISOR","department\_role":"USPS\_SUPERVISOR","department\_id":1,"employee\_id":"EMP002"}

D:\>

