DATABASE SYSTEMS – FINAL PROJECT

Software Development

TEAM: THE BIG DATA WHISPERERS

MEMBERS: Blessing Austin-Gabriel, Cristian C Noriega M, Isabele Bittencourt and Mohamed Hussein

GITHUB: <https://github.com/noriegamonsc1/1.ProjectDBaseSystemsFall2023>

YOUTUBE VIDEO: <https://www.youtube.com/watch?v=P8-y-l8rEkw>

DOCUMENTATION

**app.py**

**DB CONNECTION**

As can be seen in the screenshots, we use Flask to create an instance called app.py and then we use sqlite to connect to our database. We set the sqlachemy database URI as “sqlite:///TaxDatabaseProjectFall2023.db”. An integration is created with the app and the DB. The database is created, and we make the DB and migration object from the app.

Next, we create the taxation model with 6 columns, with tax\_id as the primary key. We create an html route to display the home page of the app.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

**APIs & CONTROLLERS:**

ENDPOINT ‘/’ (controller ONLY)

METHOD: GET

ENDPOINT ‘/fetch/TaxRecords/<dueDate>’

METHOD: POST

ENDPOINT ‘/add-record’

METHOD: POST

ENDPOINT: ‘/update/<int:tax\_id>’

METHOD: POST

ENDPOINT: ‘delete/<int:tax\_id>’

**DB SCHEMA**

A screenshot of a computer

Description automatically generated

**script.js**

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

**index.html**

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated