**FINAL PROJECT**

**Ashwini Dasanakoppa**

**1. Overview:**

This Flask application serves as a web-based tax payment management system. It allows users to manage payments for different companies, including submitting, updating, retrieving, and deleting payment records. The application integrates with an SQLite database to store and manage payment data.

**2. Features:**

1. Database Management:
   * Uses SQLite as the backend database (tax\_database.db).
   * Contains a payments table with columns: id, company, amount, payment\_date, status, due\_date, and tax\_rate.
2. Endpoints:
   * /submit: Handles form submission for adding new payment records.
   * /: Displays the index page showing all payment records and due dates.
   * /insert: Accepts JSON data to insert new payment records.
   * /summary: Generates a summary report for a selected due date, including total amounts, tax rates, and tax due.
   * /update: Updates existing payment records.
   * /delete: Deletes payment records.
3. Templates and Filters:
   * format\_date\_mmddyyyy(date\_str): Formats dates from YYYY-MM-DD to MM/DD/YYYY.
   * format\_date(value): Formats date into MM/DD/YYYY.
   * datetimeformat(value): Custom date/time formatting.

**3. Database Structure:**

* Payments Table:
  + id: Primary Key, auto incremented.
  + company: Company name (string).
  + amount: Payment amount (float).
  + payment\_date: Date of payment (string, YYYY-MM-DD).
  + status: Payment status (string).
  + due\_date: Due date for payment (string, MM/DD/YYYY).
  + tax\_rate: Tax rate associated with the payment (float).

**4. Functionality:**

* Create, Read, Update, Delete (CRUD) operations are supported.
* Allows management of payments based on due dates.
* Summary view includes financial data like total amounts and tax due based on the tax rate.

**5. CRUD Operations**

* **Create**:
  + /submit route to insert new payment records into the database.
  + Data includes company, amount, payment\_date, status, due\_date, and tax\_rate.
* **Read**:
  + / route displays all payment records.
  + /summary dynamically generates a summary based on the due\_date.
* **Update**:
  + /update route to modify existing payment records.
  + Data is retrieved from a form and updated in the database.
* **Delete**:
  + /delete route to remove payment records by their id.

**6. Controller and UI Features**

* Controller Logic:
  + Flask routes handle HTTP requests and responses.
  + Route handlers perform database operations and return rendered templates or JSON data.
* UI Features:
  + Dynamic form fields for inserting, editing, and deleting records.
  + Data formatting using custom filters like format\_date and datetimeformat.
  + Interactive summary generation using JavaScript for real-time updates.
  + Modal popups for edit operations and confirmation dialogues for delete operations.

**7. Challenges**

* Managing large datasets, optimizing database queries, ensuring data security, and handling user input validations.

**8. Potential Enhancements:**

* Authentication: Adding user authentication and authorization for secure access.
* Error Handling: Implementing more robust error handling for database operations.
* Pagination: Adding pagination for managing large datasets.
* Export: Feature to export payment data as CSV or other formats.

**9. Table Design Works**

* **Primary Key** ensures uniqueness across records.
* **Indexes** optimize search and filtering operations, improving query performance.
* The use of date formatting and calculations helps in dynamically generating tax breakdowns and summaries.

**10. Running the Application:**

* To run the application locally:
* python app.py

The application will run on http://localhost:5000/.

**Database Connection:**

**A black screen with text

Description automatically generated**

This Flask application initializes a web server and connects to a SQLite database named 'tax\_database.db'. It imports necessary modules and sets up a basic structure for handling requests, rendering templates, and managing database interactions.

**SQL SCHEMA:**

**A computer screen with text and images

Description automatically generated**

**Controllers and Endpoints:**

**A computer screen with text on it

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**A computer screen with text on it

Description automatically generated**

**A computer screen shot of text

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**A computer code on a black background

Description automatically generated**

**API’s:**

1. **POST '/submit':**

Description: Submits payment data via a form submission.

Controller: submit() function.

Method: POST**.**

1. **GET '/':**

Description: Renders the main page of the application, displaying payment records and a form for submitting new payments.

Controller: index() function.

Method: GET.

1. **POST '/insert':**

Description: Inserts a new payment record via an AJAX request.

Controller: insert\_record() function.

Method: POST.

1. **GET '/summary':**

Description: Fetches payment summary data for a specific due date.

Controller: summary() function.

Method: GET.

1. **POST '/update':**

Description: Updates an existing payment record.

Controller: update() function.

Method: POST.

1. **DELETE '/delete':**

Description: Deletes an existing payment record.

Controller: delete() function.

Method: DELETE.

**SCREENSHOTS**

**Form to enter the data and save:**

**A screenshot of a tax form

Description automatically generated**

**Dynamic Payment Summary:**

**A screenshot of a payment report

Description automatically generated**

## **Edit Record:**

Top of Form

**A screenshot of a computer

Description automatically generated**

**Delete the data:**

**After Clicking on the delete button a pop up will a appear to confirm the operation:**

**A screenshot of a computer

Description automatically generated**

**Github link:** <https://github.com/Ashwini508/Tax-Tracking-System>

**Youtube link:** [**https://youtu.be/NQNfQuNVvUA**](https://youtu.be/NQNfQuNVvUA)