**Project Title:** Pathway Exploration and Analysis Tool

**Tool Background**  
Understanding metabolic and regulatory pathways is critical for exploring the functional landscape within bacterial genomes. The Pathway Exploration and Analysis Tool aims to provide bioinformatics students and researchers with a focused, intuitive interface to browse, filter, and examine pathway information from a bacterial genome dataset. By centering on pathway data, the tool simplifies the analysis of complex metabolic and regulatory networks, helping users to quickly identify key pathways of interest, their general functions, and related metadata. Users can perform searches by pathway name or functionality, filter pathways based on certain criteria and review descriptive and functional summaries that enhance their understanding of cellular processes.

**Tool Functionality**

1. **Pathway Search**
   * **Basic Search:** Users can search for pathways by entering pathway IDs, pathway names, or keywords related to their functional roles (e.g., "glycolysis" or "DNA repair pathway").
   * **Result Filtering:** Results can be refined based on functional attributes. Users can, for example, focus on pathways associated with carbohydrate metabolism or oxidative stress response.
   * **Search Suggestions:** The tool offers suggestions as users’ type, drawn from available pathway names and descriptions in the database.
2. **Pathway Overview and Analysis**
   * **Pathway Listings:** Users can access a list of all available pathways, each displaying the pathway name and a brief functional summary.
   * **Detailed Pathway Descriptions:** By selecting a pathway, users can view additional descriptive text providing context about the pathway's metabolic role and biological significance.
   * **Functional Annotation Filtering:** Within the pathway details, users can filter by certain attributes highlighted in the pathway descriptions, focusing on specific aspects of the pathway’s function or relevance.
3. **Pathway Details**
   * **In-depth View:** Each pathway includes comprehensive descriptive information. While originally associated gene data was available, the current project scope focuses on pathway metadata such as name, description, and a functional summary.
   * **Contextual Information:** Where appropriate, pathways may include notes on their importance in cellular metabolism, precursor metabolites they produce, or their involvement in regulatory networks.
4. **Basic Visualization (Optional Future Integration)**
   * Although the current scope focuses on pathways, the tool design allows potential expansion for a future basic visualization of pathways or their related genes on a simplified chromosome map. This remains a future extension rather than a core feature of the current project.
5. **Interactive Interface**
   * **Asynchronous Loading:** Searches and filtering are managed asynchronously, enhancing the user experience by providing immediate feedback without page reloads.
   * **Expandable Sections:** Pathway descriptions and functional summaries can be expanded or collapsed to present information at varying levels of detail.
   * **User-Friendly Design:** The interface emphasizes simplicity and clarity, enabling users with minimal technical background to navigate easily.

**Technical Implementation**

1. **Database Design (MySQL)**
   * **Schema Structure:**
     + **Pathway Table:** Stores pathway names, descriptions, and functional summaries. While the original schema also included a Gene table and a pathway\_id foreign key, this project’s core now centers on the Pathway table. If the Gene table still exists, it is not the primary focus.

Example schema for Pathway:

sql

CREATE TABLE Pathway (

pathway\_id INT AUTO\_INCREMENT PRIMARY KEY,

pathway\_name VARCHAR(255) NOT NULL,

description TEXT,

class TEXT,

);

* + **Indexed Fields:** The pathway\_name field is indexed to speed up keyword searches.

1. **Backend Logic (Python CGI)**
   * **Database Connection:** The backend uses mysql.connector to connect Python CGI scripts to the MySQL database. Each script handles a specific action, such as searching for pathways or retrieving detailed descriptions.
   * **Query Optimization:** SQL queries focus on retrieving only necessary fields. A pathway search query, for example, returns pathway\_id, pathway\_name, and functional\_summary, loading full descriptions only on-demand.
   * **JSON Output Formatting:** The CGI scripts output data as JSON, enabling the front-end JavaScript to dynamically load and display pathway data. This separation of concerns improves performance and user experience.
2. **User Interface (HTML5, CSS, JavaScript, JQuery)**
   * **HTML Templates:** Structured HTML templates for displaying pathway lists, search results, and detailed descriptions help maintain clean separation between logic and presentation.
   * **Asynchronous Data Loading with JQuery:** JQuery handles asynchronous requests, letting the user refine searches, filter results, or select a pathway without reloading the page.
   * **CSS Styling and Layout:** CSS ensures a clear, readable layout, emphasizing the pathway data. Sections of the UI can be collapsed or expanded, providing a flexible, user-centric experience.
   * **Interactive Elements:** JavaScript provides the dynamic behavior for search suggestions, expandable pathway descriptions, and immediate feedback on filtering actions.
3. **Code Organization**
   * **Separation of Concerns:**
     + **Backend Scripts:** Each Python CGI script handles a discrete function, such as retrieving pathway lists or updating search filters.
     + **Frontend Files:** HTML templates, JavaScript code, and CSS stylesheets are stored separately for clarity and maintainability.
   * **Documentation:**
     + **README:** Includes setup instructions, database schema details, and usage guidelines.
     + **Inline Comments:** Each script includes comments that explain functionality, aiding future maintenance and updates.

**Technical Justification**

1. **Database Design**:  
   A single Pathway table efficiently stores and retrieves pathway-related data. By indexing pathway\_name, the tool supports fast, responsive searches.
2. **Modularity and Content Separation**:  
   Using separate CGI scripts, HTML templates, CSS, and JavaScript files enhances maintainability. The code remains clean, and adding new features (like integrating gene data in the future) is straightforward.
3. **Usability and Future Extensions**:  
   The current project emphasizes pathways, but the design accommodates future expansions. For instance, if I decide later to integrate gene data again, I can easily reintroduce the Gene table and link it via pathway\_id. The existing structure and modular codebase make such extensions manageable.

**Expected Deliverables**

1. **Deployed Tool on Class Server**:  
   The tool, focusing on pathways, will be deployed, allowing users to search and view pathway information as described.
2. **Code Submission**:  
   A GitHub repository will include CGI scripts, HTML templates, JavaScript, CSS, and MySQL setup instructions (including the Pathway table creation script).
3. **Documentation (README)**:  
   The README file will provide setup instructions, usage guidelines, and details of the file structure, along with sample data for testing pathway searches and filters.

**Future Extensions**

* Integrating gene data at a later stage to link pathways to specific genes, enabling pathway-to-gene mapping and visualization.
* Enhanced pathway visualization using interactive charts or maps.
* Cross-referencing with external databases for enriched pathway information and broader biological insights.