**Project Title: Gene Annotation and Pathway Analysis Tool**

**Project Overview**

The Gene Annotation and Pathway Analysis Tool is designed to provide users with an intuitive way to search, view, and analyze gene annotations and pathway information in a bacterial genome. Users can search for genes by ID or keyword, view detailed gene annotations, and filter results by specific pathways or gene functions. The tool uses a MySQL database to store and organize gene and pathway information, with a web interface powered by Python CGI scripts, HTML, JavaScript, and JQuery for interactivity.

**Target Users**

This tool is geared towards bioinformatics students and researchers who want an accessible interface for exploring gene and pathway data. The additional pathway filtering and gene visualization make it more informative without overwhelming users with advanced features.

**Project Functionality**

1. **Gene Search**:
   * Users can search for genes by **gene ID**, **name**, or **keywords** (e.g., functions or pathway names).
   * **Result Filtering**: Users can apply filters to search results based on pathway information or gene function (e.g., selecting genes involved in a specific metabolic pathway).
2. **Pathway and Functional Analysis**:
   * The tool displays genes categorized by pathways (e.g., metabolic pathways), allowing users to explore genes related to specific cellular functions.
   * Users can view a list of pathways with clickable entries to see genes associated with each pathway.
3. **Gene Details**:
   * Each gene entry includes detailed information such as start and end positions, gene sequence, and description of function.
   * Users can expand each gene entry to view additional details, including associated pathways or interactions.
4. **Basic Visualization**:
   * Gene start and end positions can be displayed on a simple, linear chromosome map to give users a sense of gene location.
   * This visualization will use basic HTML/CSS and JavaScript to create a basic chromosome map and gene markers, suitable for users who want a visual representation.
5. **Interactive Interface**:
   * The interface supports asynchronous loading for searches and filtering, allowing users to interact without page reloads.
   * Users can expand or collapse sections (like pathway details and gene annotations) to customize their view.

**Technical Implementation**

**1. Database Design (MySQL)**

The MySQL database will consist of two main tables:

* **Gene Table**: Stores gene attributes, including ID, name, start and end positions, function, sequence, and pathway associations.
* **Pathway Table**: Stores pathway names and descriptions, with foreign key references to associated genes in the Gene table.

This schema allows for efficient retrieval of gene information with pathway-specific filtering.

**2. Backend Logic (Python CGI)**

* **Database Connection**: Use mysql.connector to handle connections between CGI scripts and MySQL. Each CGI script handles a distinct task (e.g., search, filter, retrieve gene details).
* **Search and Filter Functionality**: The CGI scripts parse user inputs to build SQL queries for gene ID, keywords, and pathway filters. Each query retrieves only the necessary fields, minimizing load.
* **Data Formatting and JSON Output**: Data is formatted for HTML and JSON output, allowing JavaScript to dynamically load data on the front end.

**3. User Interface (HTML5, CSS, JavaScript, JQuery)**

* **HTML Templates**: HTML is structured with templates for search, results, and detailed views, keeping the CGI code and HTML presentation separate.
* **JavaScript and JQuery**: JQuery handles asynchronous search and filtering requests, while basic JavaScript updates the chromosome map based on gene location.
* **CSS Styling**: CSS styling will emphasize readability and clean layout, with gene visualization elements styled to keep the interface intuitive.

**Code Organization**

1. **Separation of Concerns**:
   * CGI scripts will handle the backend logic and avoid embedding HTML directly within Python code.
   * HTML templates, JavaScript, and CSS files will be stored in separate folders, making it easy to update or modify the front-end independently.
2. **Documentation**:
   * A README file will provide setup instructions, descriptions of each script and file, and usage guidelines.
   * Inline comments in scripts will help clarify the purpose of each section for maintainability.

**Technical Justification**

1. **Database Design**:
   * A relational MySQL schema is used to organize gene and pathway information efficiently. This allows complex queries, like pathway-specific filters or keyword searches, to be executed with minimal impact on performance.
2. **Modularity and Content Separation**:
   * Separate HTML templates, JavaScript, and CSS files enhance modularity and readability, which is critical for an intermediate-level project.
   * Using JQuery for asynchronous searches minimizes server load and provides a smooth user experience.
3. **Enhanced Usability through Visualization**:
   * Simple gene visualization provides spatial context to gene annotations without overwhelming users. CSS and JavaScript handle this efficiently, keeping the tool lightweight.

**Expected Deliverables**

1. **Deployed Tool on Class Server**:
   * Fully deployed on the class server with search, filtering, and pathway analysis features available as described.
2. **Code Submission**:
   * Tarball of all project files, including CGI scripts, HTML templates, JavaScript, CSS, and MySQL setup scripts.
3. **Documentation (README)**:
   * Clear instructions for setup, usage, and explanations of file roles. A step-by-step guide will be included for loading sample data.

**Timeline**

* **November 15**: Submit project proposal.
* **November 16 – December 15**: Develop and test each feature iteratively, with feedback adjustments.
* **December 16**: Final deployment and submission of project code on Canvas.

**Future Extensions**

Future features could include support for additional bacterial genomes, more complex visualization (such as interactive graphs or pathway maps), and the ability to cross-reference other databases. These additions could enhance usability but are beyond the scope of the initial project.

This project strikes a balance between functionality and simplicity, giving users an enriched experience without overwhelming technical demands. Let me know if you need additional details or suggestions!

Top of Form

Bottom of Form