GeneAlchemyApp - README

Team- NEMOS

1. Overview

GeneAlchemyApp is a Java-based desktop application built using the **Swing** GUI library. The application features a user authentication system that allows users to sign up and log in, with their credentials securely stored and validated against a MySQL database. It features a user-friendly interface with a **CardLayout** that switches between the login and sign-up screens, providing a seamless user experience.

The main functionality of the application is:

- User registration (sign-up) and login.
- Storing usernames and passwords securely in a MySQL database.
- Switching between the login and sign-up forms using CardLayout.

2. Features

Login Page

- Allows users to enter their **username** and **password** to authenticate.
- On successful login, opens another application window (BioAppGUI), though the BioAppGUI class itself is not yet implemented in this code.

Sign-Up Page

- Allows new users to **create an account** by providing a **username**, **password**, and confirming the password.
- On successful registration, the new user data is stored in the MySQL database.

Database Integration

- Uses **JDBC** (Java Database Connectivity) to interact with a MySQL database.
- User credentials are securely inserted and validated against the database.

GUI with CardLayout

- The **login** and **sign-up** pages are handled using **CardLayout**, which allows for easy switching between the two forms within the same window.

3. System Requirements

Hardware

- **Processor**: Any processor capable of running Java applications.
- **RAM**: At least 2 GB of RAM (recommended 4 GB for smooth performance).
- **Disk Space**: At least 100 MB for application files and database.

Software

- **Java Development Kit (JDK)** version 8 or higher.
- **MySQL Database Server** version 5.x or higher.

- **JDBC Driver for MySQL** (typically mysql-connector-java).

4. Setup Instructions

1. Setting Up MySQL Database

Before running the application, you need to set up a MySQL database to store user information.

Install MySQL:

- Download and install MySQL from the official website.
- Make sure MySQL server is running.
- Create Database and Table:
- Open MySQL Workbench or any MySQL client and connect to your server.

Create a new database called GeneAlchemyDB:

```
CREATE DATABASE GeneAlchemyDB;
```

Create a users table in the GeneAlchemyDB database with the following SQL:

```
CREATE TABLE users (

   id INT AUTO_INCREMENT PRIMARY KEY,

   username VARCHAR(255) UNIQUE NOT NULL,

   password VARCHAR(255) NOT NULL

);
```

- Configure Database Credentials:
- In the Java code, update the database connection string to match your MySQL server's username and password:
  `conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/GeneAlchemyDB", "root", "your_password");`

2. Java Setup

1. **Install JDK**:
   - Ensure that you have JDK 8 or higher installed on your system.
   - You can download it from the Oracle JDK website.
2. **Install an IDE**:
   - You can use an IDE like **IntelliJ IDEA**, **Eclipse**, or **NetBeans** to compile and run Java applications.
   - Alternatively, you can use the terminal/command line to compile and run Java files.

3. Running the Application

1. **Compile the Java Code**:

If using the terminal, navigate to the directory containing your .java file and compile it using:

```
javac GeneAlchemyApp.java
```

2. **Run the Application**:

After compiling the code, run the application using:

```
java GeneAlchemyApp
```

This will launch the GUI, displaying the login page by default.

5. User Flow

Login Process

- When the application starts, the **Login Page** is shown.
- The user enters their **username** and **password** and clicks the **Login** button.
- The credentials are validated against the database.
- If successful, the application opens another window (currently, the BioAppGUI class is empty but should be implemented).
- If login fails, an error message is displayed.

Sign-Up Process

- From the login page, the user can click the **Sign Up** button to navigate to the **Sign-Up Page**.
- The user enters a **username**, **password**, and **confirm password** and submits the form by clicking the **Submit** button.
- If the passwords match, the user is added to the database, and the login page is shown.
- If there's an error or passwords don't match, an error message is displayed.

6. Database Schema

The MySQL database schema is quite simple. It consists of a single table:

Column Name	**Data Type**	**Description**
id	INT (AUTO INCREMENT)	Unique ID for each user
username	VARCHAR(255)	Unique ID for each user
password	VARCHAR(255)	The user's password

7. Code Overview

Key Components

- **GeneAlchemyApp Class**:
  - This is the main entry point of the application. It initializes the GUI components and handles the logic for switching between login and sign-up screens using **CardLayout**.
  - It also manages the database connection and validates user credentials.
- **Database Connection**:
  - The code establishes a **JDBC connection** to the MySQL database using the DriverManager.getConnection() method.
- **Event Handling**:
  - **Login Button**: When the **Login** button is clicked, it validates the user's credentials against the database using the validateLogin() method.
  - **Sign Up Button**: When the **Sign Up** button is clicked, it switches to the sign-up form using cardLayout.show(cardPanel, "SignUp").
  - **Sign-Up Submit Button**: This button checks if the passwords match and inserts the user into the database.

8. Troubleshooting

MySQL Connection Issues

- **Error**: "Unable to connect to MySQL server."
  - Ensure MySQL is running and accessible.
  - Check if the correct username, password, and database name are provided in the connection string.

GUI Issues

- **Error**: "Swing components not showing."
  - Make sure that frame.setVisible(true) is called after adding all the components.
  - Ensure that your IDE or terminal is configured correctly to handle Swing applications.

BioAppGUI - README

GeneAlchemy is a user-friendly Java-based application designed to assist researchers, students, and enthusiasts in exploring and analyzing DNA and RNA sequences. The tool provides a suite of utilities to calculate GC content, transcribe DNA to RNA, translate RNA to proteins, and much more.

Features

About Page

The About Page introduces GeneAlchemy and its purpose. It also provides a fun, interactive feature:

- **Fun Fact Button:** Displays a random fact about DNA.

Utilities Page

GeneAlchemy offers the following DNA and RNA utilities:

DNA Functions

1. **GC Content Calculator -** Calculates the GC content percentage in a DNA sequence.
2. **Complement Finder -** Determines the complementary DNA strand (A-T, T-A, C-G, G-C).
3. **Transcribe to RNA -** Converts DNA to RNA by replacing thymine (T) with uracil (U).
4. **Motif Finder -** Identifies all occurrences of a specific motif in the DNA sequence.
5. **Palindrome Finder -** Finds palindromic sequences in the DNA.
6. **Nucleotide Counter -** Counts occurrences of A, T, C, G nucleotides.

RNA Functions

1. **GC Content Calculator -** Calculates the GC content percentage in an RNA sequence.
2. **Transcribe to Protein -** Translates RNA into a protein sequence using codons.
3. **Motif Finder -** Identifies motifs in RNA sequences.
4. **Palindrome Finder -** Detects palindromic sequences in RNA.
5. **Nucleotide Counter -** Counts occurrences of A, U, C, G nucleotides.

Installation

Prerequisites

- Java Development Kit (JDK) 8 or higher
- An IDE or text editor for Java (e.g., IntelliJ IDEA, Eclipse, or Visual Studio Code)
- A Java-compatible operating system (Windows, macOS, Linux)

Steps

Clone the repository:

```
git clone https://github.com/your-username/GeneAlchemy.git
cd GeneAlchemy
```

Compile the Java code:

```
javac BioAppGUI.java
```

Run the application:

```
java BioAppGUI
```

Usage

1. Launch the application.
2. Navigate between the **About Page** and **Utilities Page** using the provided interface.
3. Input DNA/RNA sequences to analyze using the tools on the Utilities Page.
4. Explore the fun facts and gain insights about DNA through the Fun Fact feature.

File Structure

- `BioAppGUI.java`: Main application file containing the user interface and logic.
- **Images**: Folder containing assets such as the `aboutus.jpeg` image.