Experiment 6: Servlet Controller App

Objective

Build a servlet to link the frontend shopping cart with the database backend.

Technologies Used

- Java Servlet
- HTML, JDBC

Features

- Handle form submissions
- Database operations via servlet

Steps to Execute

1. Set up Tomcat.
2. Place app in `webapps/yourapp/`.
3. Add DB config in `ServletController.java`.
4. Access from browser.

Folder Contents

- `ServletController.java`
- `web.xml`

Experiment 6: Servlet-Based Controller for Shopping Cart

Folder Name (from image): `Experiment-06_Servlet-Based-Controller`

Description (from document): "Design a controller with servlet that provides the interaction with application developed in experiment 1 and the database created in experiment 5."

`README.md` **for Experiment 6:**

```
# Experiment 6: Servlet-Based Controller for Shopping Cart Application

This project integrates the frontend of the shopping cart application (from
Experiment 1/2/3) with the backend database operations (from Experiment 5) using a
**Servlet-based controller**. The goal is to establish a full-stack flow where
user interactions on the web pages (e.g., registration, login, adding items) are
handled by Servlets, which then interact with the database using JDBC. This
demonstrates the Model-View-Controller (MVC) pattern in a basic Servlet
application.

## Features
```

```
* **Servlet as Controller**: A central `FrontControllerServlet` (or multiple specific servlets) handles incoming HTTP requests from the web pages.
* **Request Handling**: Processes form submissions (Registration, Login) and navigates between pages.
* **Database Interaction**: The Servlets invoke methods that perform CRUD operations on the `STUDENTS` table (and potentially `PRODUCTS`, `CART` tables) using JDBC, connecting to the database (Oracle/MySQL) as developed in Experiment 5.
* **Data Flow**:
    * Frontend (HTML/JSP) sends data to Servlets.
    * Servlets process data, interact with the database.
    * Servlets forward/redirect to appropriate JSP pages (Views) with processed data or messages.
* **Integration with Frontend**: Designed to work with the existing HTML/CSS/JS/Bootstrap files from previous experiments (1, 2, 3) to provide a dynamic user experience.
* **Error Handling**: Basic error handling for database issues or invalid inputs within the servlet.

## Technologies Used

* Java SE
* Java Servlets API
* JSP (JavaServer Pages) for Views
* JDBC API (for database interaction)
* Oracle Database OR MySQL Database (as used in Experiment 5)
* Oracle JDBC Driver OR MySQL Connector/J
* HTML5, CSS3, JavaScript (from previous frontend experiments)
* Apache Tomcat (or any Servlet Container)

## Prerequisites

1. **JDK and Apache Tomcat**: Installed and configured.
2. **Database Setup (from Experiment 5)**:
    * A running Oracle or MySQL database instance.
    * The `STUDENTS` table (and any other necessary tables like `PRODUCTS`, `CARTS`) created in your database.
    * A database user with appropriate privileges.
3. **JDBC Driver**: The correct `ojdbcX.jar` or `mysql-connector-java-X.X.X.jar` must be available to the web application (e.g., in `WEB-INF/lib`).
4. **Frontend Pages**: The HTML/JSP pages for Registration, Login, Catalog, Cart (from Experiments 1, 2, 3) are assumed to be present and structured to send requests to the servlet.

## Setup and Running

1. **Clone the Repository (or create manually):**
    ```bash
    git clone [https://github.com/your-username/Experiment-06_Servlet-Based-Controller.git](https://github.com/your-username/Experiment-06_Servlet-Based-Controller.git)
    cd Experiment-06_Servlet-Based-Controller
    ```
```

```
Experiment-06_Servlet-Based-Controller/images/2.png
Experiment-06_Servlet-Based-Controller/images/1.png
Experiment-06_Servlet-Based-Controller/images/servlet.png

2.  **Copy Frontend Files:**
    * Copy the `html`, `css`, `js` files (and any `images/` or `lib/` for
Bootstrap) from your `Experiment-01`, `Experiment-02`, and `Experiment-03`
projects into the `src/main/webapp/` directory of this project. Ensure form
`action` attributes in your HTML/JSP point to the correct servlet URLs (e.g.,
`action="register"`, `action="login"`).

3.  **Add JDBC Driver to Web Application:**
    * Create a `WEB-INF/lib/` directory inside `src/main/webapp/` if it doesn't
exist.
    * Place your `ojdbcX.jar` (for Oracle) or `mysql-connector-java-X.X.X.jar`
(for MySQL) into this `WEB-INF/lib/` folder. This makes the driver available to
your Servlets.

4.  **Update Database Configuration:**
    * Open the Servlet class (e.g., `FrontControllerServlet.java` or
`DatabaseUtil.java` if using a separate utility class).
    * Update the database connection details (URL, user, password) to match your
setup, similar to Experiment 5.

5.  **Project Setup in IDE (e.g., IntelliJ IDEA, Eclipse):**
    * Create a new Dynamic Web Project (Eclipse) or Jakarta EE Web Application
(IntelliJ IDEA).
    * Name it (e.g., `ShoppingCartServletApp`).
    * Place the Java servlet files (e.g., `FrontControllerServlet.java`,
`UserDAO.java`) into `src/main/java/com/example/controller` (or appropriate
package).
    * Ensure `web.xml` is in `src/main/webapp/WEB-INF/`. The servlet(s) should be
mapped here or via `@WebServlet` annotations.
    * Configure your Tomcat deployment for the project (e.g., set the `Application
context` to `/ShoppingCartApp`).

6.  **Build and Deploy:**
    * Build the project within your IDE.
    * Deploy the `WAR` file or "exploded" artifact to your Tomcat server.

7.  **Run the Application:**
    * Start your Tomcat server.
    * Open your web browser and navigate to:
`http://localhost:8080/ShoppingCartApp/index.html` (replace `ShoppingCartApp` with
your actual application context).

## Project Structure

.
├── src/main/java/com/example/controller/
│   ├── FrontControllerServlet.java # Main servlet handling requests
│   └── UserDAO.java                # Example DAO for database interaction
│   └── ProductDAO.java             # (Optional) DAO for products
```

```
├── src/main/webapp/
│   ├── WEB-INF/
│   │   ├── web.xml                # Deployment descriptor
│   │   └── lib/                   # JDBC driver goes here
│   │       ├── ojdbc11.jar        # Oracle JDBC Driver (example)
│   │       └── mysql-connector-java-X.X.X.jar # MySQL Connector/J (example)
│   ├── css/                       # Copied from Exp 1/2
│   ├── js/                        # Copied from Exp 3
│   ├── registration.html          # Frontend page for registration
│   ├── login.html                 # Frontend page for login
│   ├── catalog.jsp                # Or .html for catalog display
│   ├── cart.jsp                   # Or .html for cart display
│   └── index.html                 # Main entry point
└── README.md
```