Project 2

• CMUQFlix!
• A Movie Recommendation System
Project 2 Objectives

• Set up a front-end website with PostgreSQL as the back-end

• Allow users to login, “like” movies, and get personalized movies recommendations
Three-Tier Architecture of CMUQFlix

Client Browser

Front-end Tier (presentation)

HTTP

Web Server

Middle Tier (application logic)

Tomcat

Servlets, JSPs, Java

JDBC

PostgreSQL Database

Back-end Tier (data storage)

Movies, Users, ..
**Front-End Tier: HTML Forms**

- HTML forms are a common way for
  - Getting user input at the front-end tier
  - Communicating the given input to the middle tier

- HTML forms are defined using the `FORM` tag

- `FORM` tag has three important attributes:
  - `NAME`: form name
  - `ACTION`: URI of the page to which the form is submitted
  - `METHOD`: method used to submit the form
    (Get* / Post**)

* A **Get** request gets (or retrieves) information (e.g. image, HTML page) from a server.
** A **Post** request posts (or sends) data from the client (e.g. usernames, passwords) to a server.
Front-End Tier: HTML Forms

• Three important tags used to get user input: **INPUT, SELECT, and TEXTAREA**

• **INPUT** is used to render text fields and buttons:

```html
<INPUT TYPE="text" NAME="username" VALUE="John Brown">
<INPUT TYPE="password" NAME="password" VALUE="123456">
<INPUT TYPE="reset" VALUE="Reset Fields">
<INPUT TYPE="submit" VALUE="Submit Form">
```

![HTML Form Example](image)
Back-End Tier

- PostgreSQL:
  - We already know this very well!
Middle Tier: Apache Tomcat

• **Apache Tomcat** is an **open-source web server** developed by the Apache Software Foundation (ASF)

• Apache Tomcat is a servlet container i.e. it can invoke and execute servlets and JSPs (we’ll see what these are in a bit)

• Unlike web-servers that run on a TCP port number 80, **Tomcat runs on port number 8080**
Installing Tomcat

• Login to your VM (<andrew_id>@<andrew_id>-db.qatar.cmu.local)
• Install JDK 7 (apt-get install openjdk-7-jdk)
• Install Tomcat 7 (apt-get install tomcat7 tomcat7-admin)
• Check that Tomcat is running
  • Type service tomcat7 status, or
  • Browse to <andrew_id>-db.qatar.cmu.local:8080
• Manually start/stop/restart Tomcat (service tomcat7 start/stop/restart)
• Deploy your application “cmuqflix” in Tomcat’s webapp (cp cmuqflix /var/lib/tomcat7/webapps)
• View Tomcat’s log files (for debug info.) (cd /var/log/tomcat7; cat *)
Java Servlets

• A servlet is primarily **Java code** that executes some application logic and produces **HTML output**

• A servlet can:
  • Interact with databases (using **JDBC**)
  • Generate custom HTML pages (as output)
  • Maintain session information for each client
Creating a Servlet

• A web-based servlet must:
  • Extend the `HTTPServlet` abstract class
  • Override the `doGet` and/or `doPost` functions
    (to respond to get and post HTTP requests)

• The web-server (i.e. Tomcat) transparently invokes a Service to determine the type of HTTP request made

<table>
<thead>
<tr>
<th>Methods defined in the HttpServlet Abstract Class*</th>
</tr>
</thead>
<tbody>
<tr>
<td>void doGet ( HttpServletRequest request, HttpServletResponse response )</td>
</tr>
<tr>
<td>void doPost ( HttpServletRequest request, HttpServletResponse response )</td>
</tr>
<tr>
<td>void Service( HttpServletRequest request, HttpServletResponse response )</td>
</tr>
</tbody>
</table>

* Note that all the above functions can throw ServletException and IOException
Request and Response Classes

• Class **HttpServletRequest** provides methods:
  - `getParameter`, `getParameterNames`, `getParameterValues`, `getCookies`, `getSession`

• Class **HttpServletResponse** provides methods:
  - `addCookie`, `getOutputStream`, `getWriter`, `setContentType`
Request and Response Classes

• Class **HttpServletRequest** provides methods:
  - `getParameter`, `getParameterNames`, `getParameterValues`, `getCookies`, `getSession`

• Class **HttpServletResponse** provides methods:
  - `addCookie`, `getOutputStream`, `getWriter`, `setContentType`
Integrating a Servlet with Tomcat

- Define the new servlet (e.g., `newservlet.java`) under `cmuqflix/WEB-INF/src`

- Compile `newservlet.java`
  ```bash
  javac -cp /usr/share/tomcat7/lib/servlet-api.jar newservlet.java
  ```

- Copy `newservlet.class` to `cmuqflix/WEB-INF/classes`

- Define the new servlet in `cmuqflix/WEB-INF/web.xml`