Task 1

- How did you use connection pooling?
  Define two data sources (one for read only, one for write only) that enables connection pooling in context.xml; in servlets use context lookup to get connection from the pool.

  - File name, line numbers as in Github
    Read pool:
    - fabflix/src/CheckoutServlet.java - Lines 65-70 (Note: I used both Read and Write connection for this Servlet)
    - fabflix/src/ConfirmationServlet.java - Lines 63-66
    - fabflix/src/DashServlet.java - Lines 74-77
    - fabflix/src/MovieSearchServlet.java - Lines 54-57
    - fabflix/src/MovieSuggestionServlet.java - Lines 57-60
    - fabflix/src/ShoppingCartServlet.java - Lines 68-71
    - fabflix/src/SingleMovieServlet.java - Lines 57-60
    - fabflix/src/SingleStarServlet.java - Lines 57-60
    - fabflix/src/DashServlet.java - Lines 68-71

    Write pool:
    - fabflix/src/CheckoutServlet.java - Lines 65-70 (See above)
    - fabflix/src/DashServlet.java - Lines 128-131, 157-160

  - Snapshots showing use in your code

```
65              Context initCtx = new InitialContext();
66              Context envCtx = (Context) initCtx.lookup("java:comp/env");
67              DataSource dsRead = (DataSource) envCtx.lookup("jdbc/movieRead");
68              DataSource dsWrite = (DataSource) envCtx.lookup("jdbc/movieWrite");
69              dbRead = dsRead.getConnection();
70              dbWrite = dsWrite.getConnection();
```

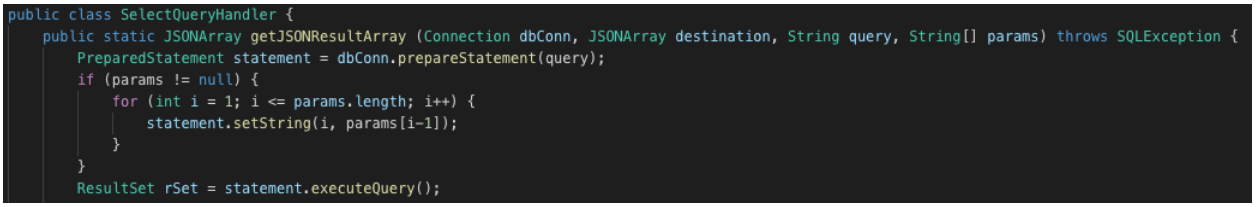
    For more snapshots please see README.md

- How did you use Prepared Statements?
  In data sources in context.xml, define parameter “cachePrepStmts=true”.
  All SQL queries with SELECT statement can be handled with “getJSONResultArray” function of “SelectQueryHandler” class, returns a JSONArray object, which uses Prepared Statements to contact with Databases.
  Please see README and fabflix/src/SelectQueryHandler.java

  - File name, line numbers as in Github
    fabflix/src/LoginServlet.java - Line 100
    fabflix/src/MovieSearchServlet.java - Lines 66, 70, 71
    fabflix/src/SingleMovieServlet.java - Lines 77, 80, 83
    fabflix/src/SingleStarServlet.java - Lines 71, 74
    fabflix/src/StaffLoginServlet.java - Line 75
    fabflix/src/DashServlet.java - Lines 81, 87, 146, 174, 207

fabflix/src/ShoppingCartServlet.java - Line 79
fabflix/src/CheckoutServlet.java - Line 91
fabflix/src/ConfirmationServlet.java - Line 83

- Snapshots showing use in your code



```
public class SelectQueryHandler {
    public static JSONArray getJSONResultArray (Connection dbConn, JSONArray destination, String query, String[] params) throws SQLException {
        PreparedStatement statement = dbConn.prepareStatement(query);
        if (params != null) {
            for (int i = 1; i <= params.length; i++) {
                statement.setString(i, params[i-1]);
            }
        }
        ResultSet rSet = statement.executeQuery();
```

For full function please see fabflix/src/SelectQueryHandler.java.
For more usage please see README file.

Task 2

- Address of AWS and Google instances

use the following public IP addresses and ports:

Plugin	IP	Opening Port
Google Cloud	35.185.3.85	80 (Tomcat not opened!)
AWS Instance 1	34.217.146.202	80, 8080 both opened
AWS Instance 2	52.24.164.192	8080
AWS Instance 3	54.188.51.12	8080

IP subject to change of Google Sheet!!!!!!!

Note: logging, HTTPS are disabled according to related permission.

- Have you verified that they are accessible? Does Fablix site get opened both on Google's 80 port and AWS' 8080 port?

  The IP addresses with URL for fabflix is posted on Google sheet. See opened port on sheet above.

- Explain how connection pooling works with two backend SQL (in your code)?

Connection Pooling is enabled in fabflix/WebContent/META-INF/context.xml, where two data sources are defined. Data Source for all Mysql Reads goes to "jdbc/movieRead", which could connect to local database of the instance that the Tomcat is in, because for read-only operations JDBC can connect to either master and slave. On the other hand, all Mysql Writes must connect to data source named "jdbc/movieWrite", which connects to the IP of the master instance.

- File name, line numbers as in Github

In fabflix/WebContent/META-INF/context.xml, “jdbc/movieRead” and “jdbc/movieWrite” are defined in lines 21-24, 26-29, respectively.

- Snapshots

```
21    <Resource name="jdbc/movieRead" auth="Container" type="javax.sql.DataSource"
22              maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="mytestuser"
23              password="mypassword" driverClassName="com.mysql.jdbc.Driver"
24              url="jdbc:mysql://localhost:3306/moviedb?autoReconnect=true&amp;useSSL=false&amp;cachePrepStmts=true"/>
25
26    <Resource name="jdbc/movieWrite" auth="Container" type="javax.sql.DataSource"
27              maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="mytestuser"
28              password="mypassword" driverClassName="com.mysql.jdbc.Driver"
29              url="jdbc:mysql://52.24.164.193:3306/moviedb?autoReconnect=true&amp;useSSL=false&amp;cachePrepStmts=true"/>
```

- How read/write requests were routed?
  See explanation above.
  - File name, line numbers as in Github
    See above.
  - Snapshots
    See context.xml snapshot.

Task 3

- Have you uploaded the log files to Github? Where is it located?
- Have you uploaded the HTML file (with all sections including analysis, written up) to Github? Where is it located?
- Have you uploaded the script to Github? Where is it located?
- Have you uploaded the WAR file and README to Github? Where is it located?

Log files of task 3 are located at test/*.txt
The script is located in test/analy.py
The WAR file is located at out/fabflix.war
The README.md file is located at the project root directory, also copied into /fabflix directory, served as the report of Project 5.
The report.pdf file, following the provided report guideline, is located at located at the project root directory, also copied into /fabflix directory.
The measurement HTML file, named jmeter_report.html, is located at located at the project root directory, also copied into /fabflix directory.
Networking is required for viewing all graphs and snapshots in README.md and jmeter_report.html.
This root directory for all files mentioned in this report is the project directory in GitHub.

I’ve also copied the script into /fabflix directory!
(Since it will be in root of .war after importing that.war file)

Ok it doesn’t work…… I copied into the same directory with the war file. Now it’s EVERYWHERE……

BTW for more explanation for script usage please reference README.md