**Task 1**

* How did you use connection pooling?

We first added the database connections in our context.xml and web.xml. In our API backend, we edited MovieListServlet, MovieAutoComplete, SingleMovieServlet, StarServlet to allow connection pooling. This was done by using a database that pointed to our master MySQL instance.

* File name, line numbers as in Github

MovieListServlet.java, line 98

MovieAutoComplete.java, line 75

SingleMovieServlet.java, line 51

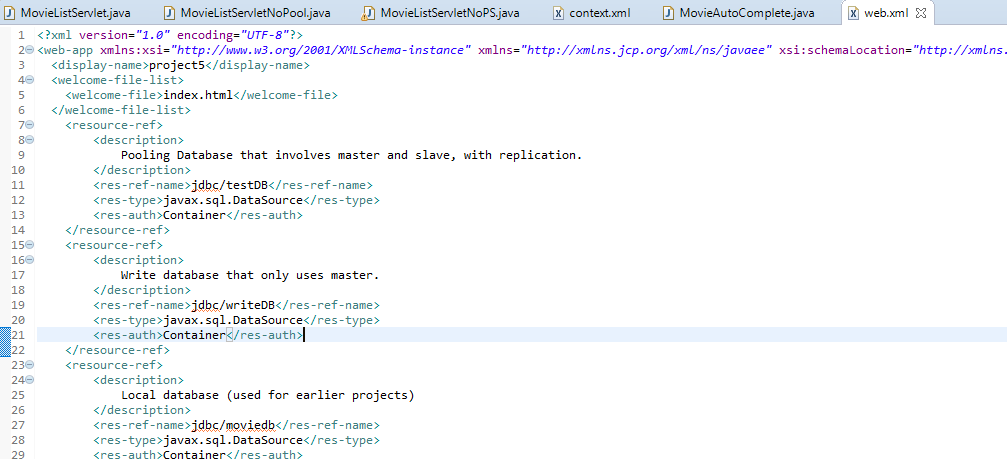
StarServlet.java, line 45

* Snapshots showing use in your code

Context.xml



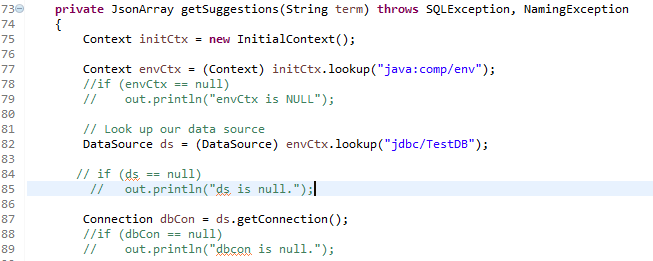
Web.xml



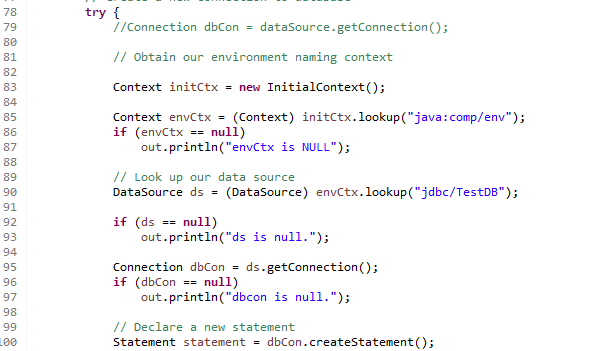
MovieListServlet



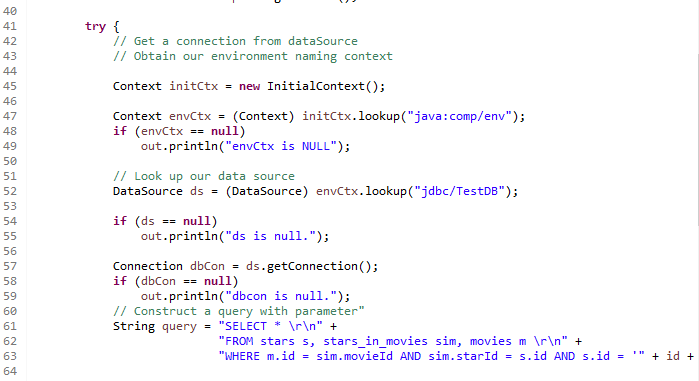
MovieAutoComplete



SingleMovieServlet



StarServlet



* How did you use Prepared Statements?

In our search related functions we used prepared statements and then updated the queries, followed by executing it completely. We saved the result and then used the information in our JSON objects. This was done in MovieListServlet.java and MovieAutoComplete.java.

* File name, line numbers as in Github

MovieListServlet.java, line 185~

MovieAutoComplete.java, line 94

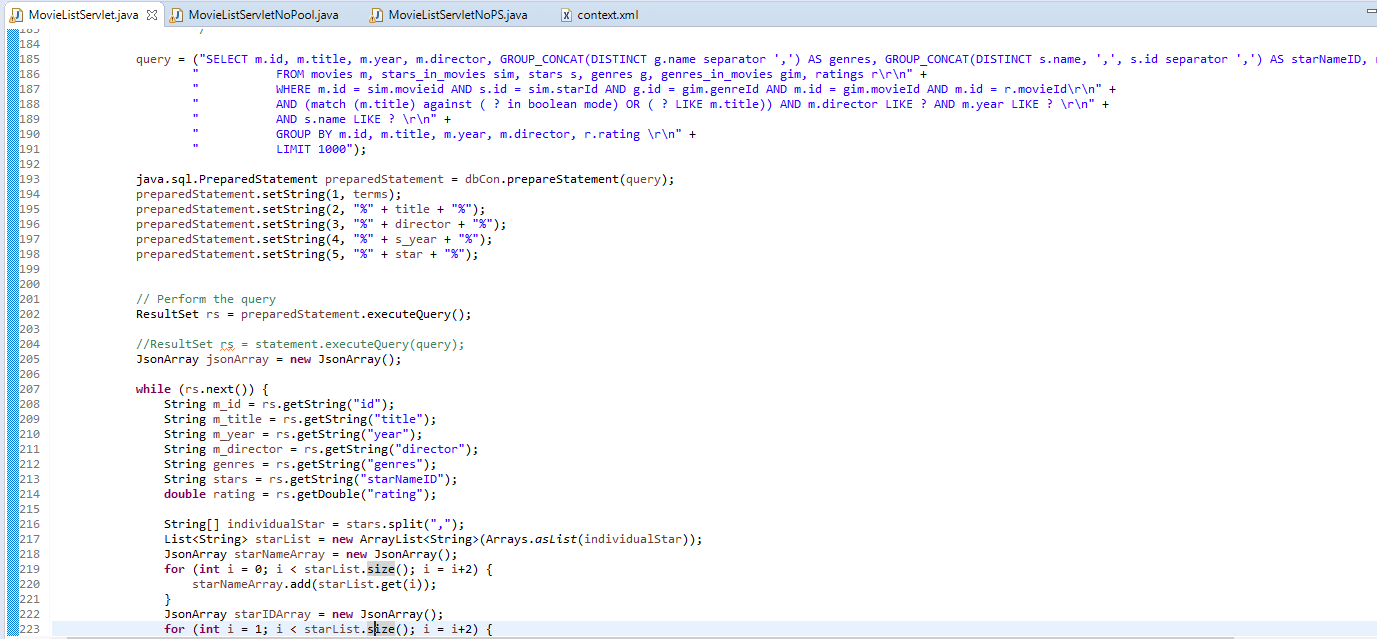
AddMovieServlet.java, line 70

AddInfoServlet.java, line 73

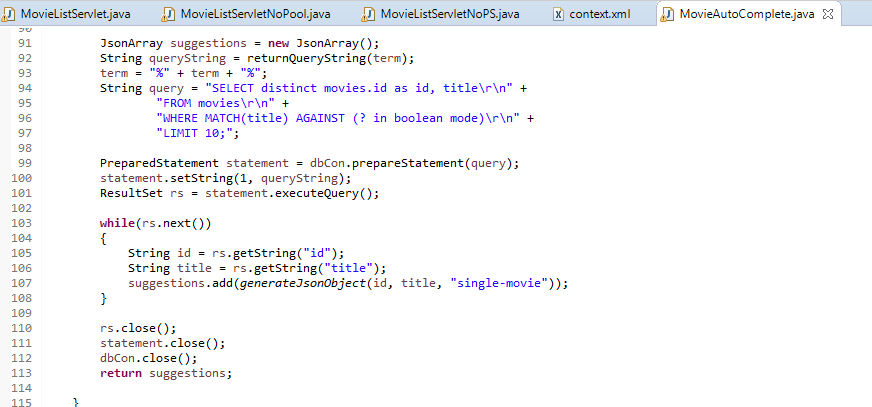
AddStarServlet.java, line 70

* Snapshots showing use in your code

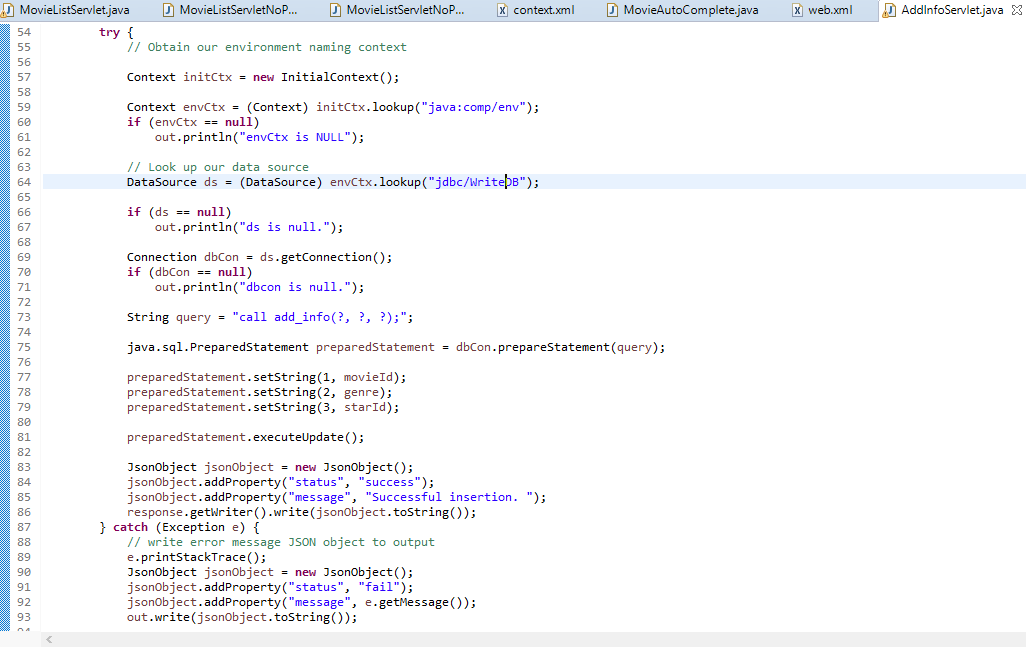
MovieListServlet



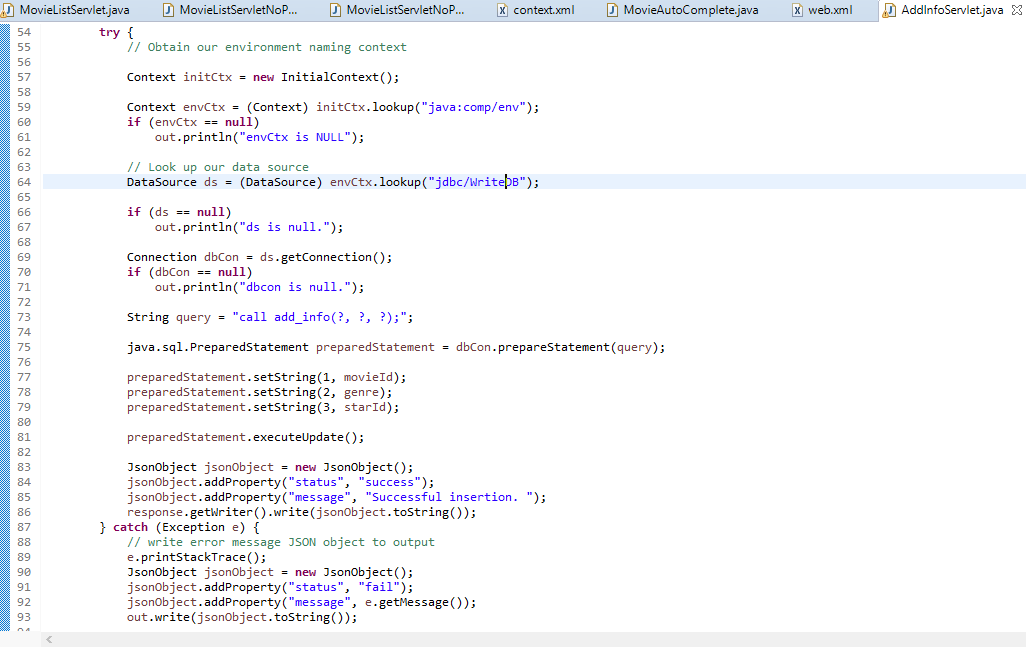
MovieAutoList



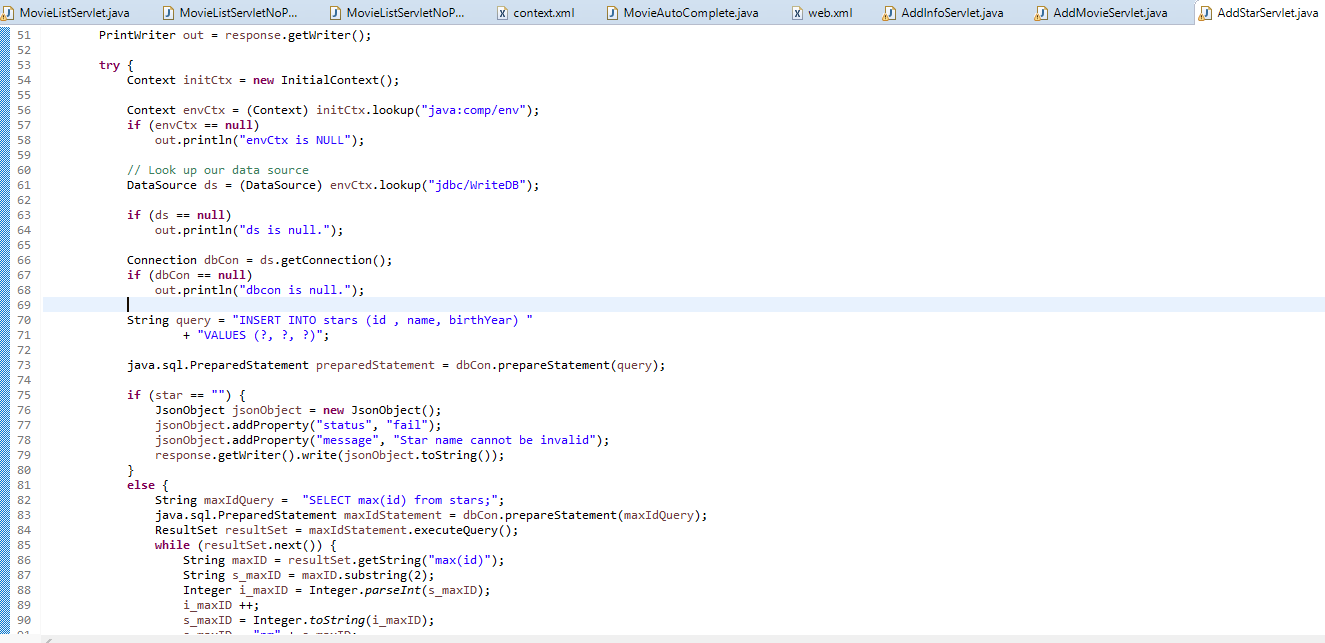
AddMovieServlet



AddInfoServlet



AddStarServlet



**Task 2**

* Address of AWS and Google instances

AWS 1 (Original):

ec2-18-220-242-149.us-east-2.compute.amazonaws.com

18.220.242.149

AWS 2 (Master):

ec2-18-221-144-12.us-east-2.compute.amazonaws.com

18.221.144.12

AWS 3 (Slave):

ec2-13-58-169-104.us-east-2.compute.amazonaws.com

13.58.169.104

Google:

[35.196.242.225](https://35.196.242.225/)

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

Yes.

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

In my context.xml I list 3 databases, one for local (original), one for a master/slave connection with replication, and a third, write only database connection to the master.

* + File name, line numbers as in Github

Context.xml

* + Snapshots



* How read/write requests were routed?

For my searches I used the context database of the master/slave. For functions that actually write to the database, I swapped them to use writeDB instead.

* + File name, line numbers as in Github

MovieListServlet.java, line 98

MovieAutoComplete.java, line 75

SingleMovieServlet.java, line 51

StarServlet.java, line 45

AddMovieServlet.java, line 70

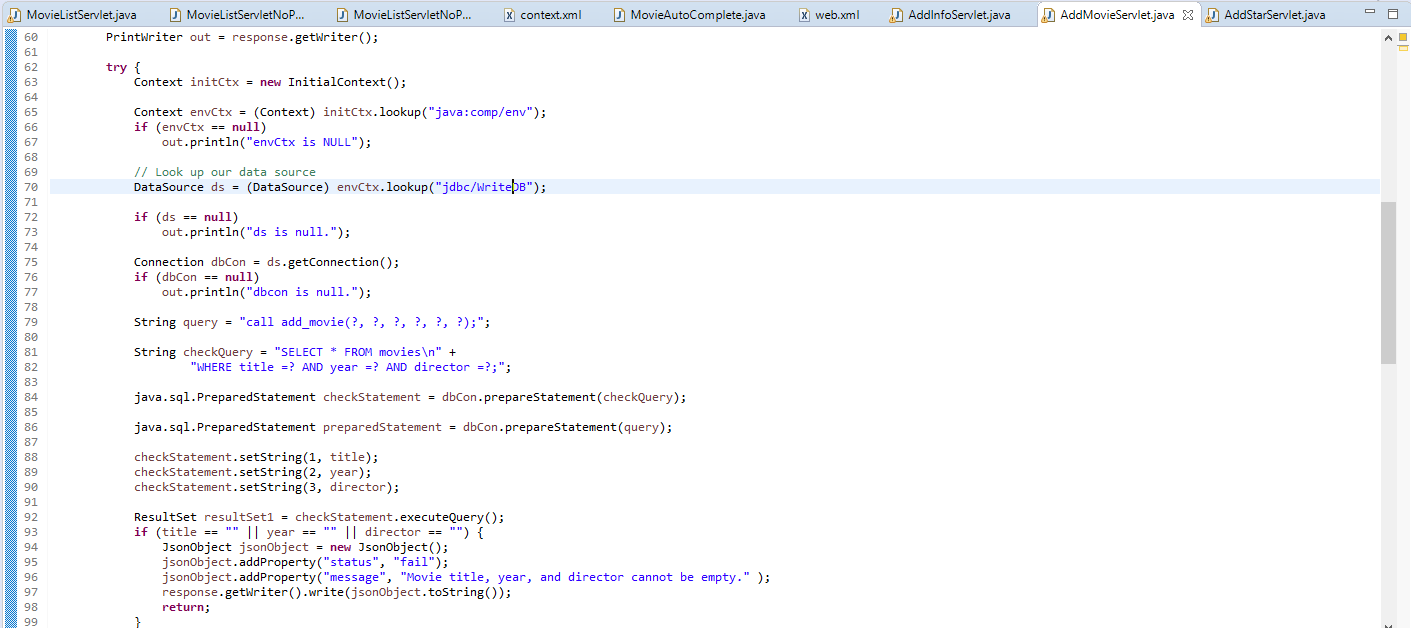
AddInfoServlet.java, line 64

AddStarServlet.java, line 61

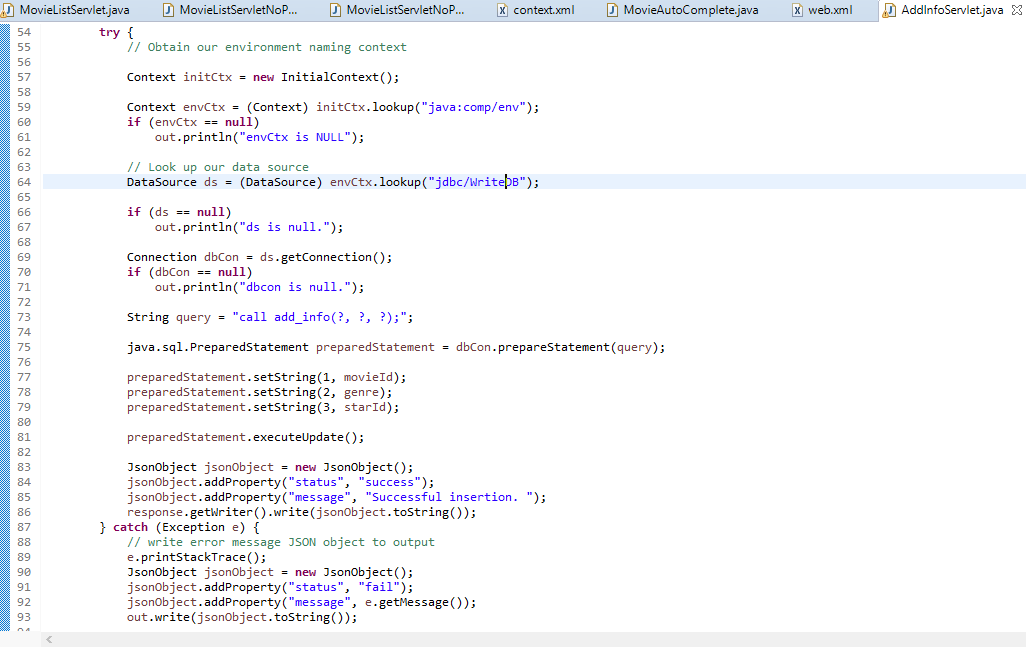
* Snapshots

Listed in task 1 for first 4.

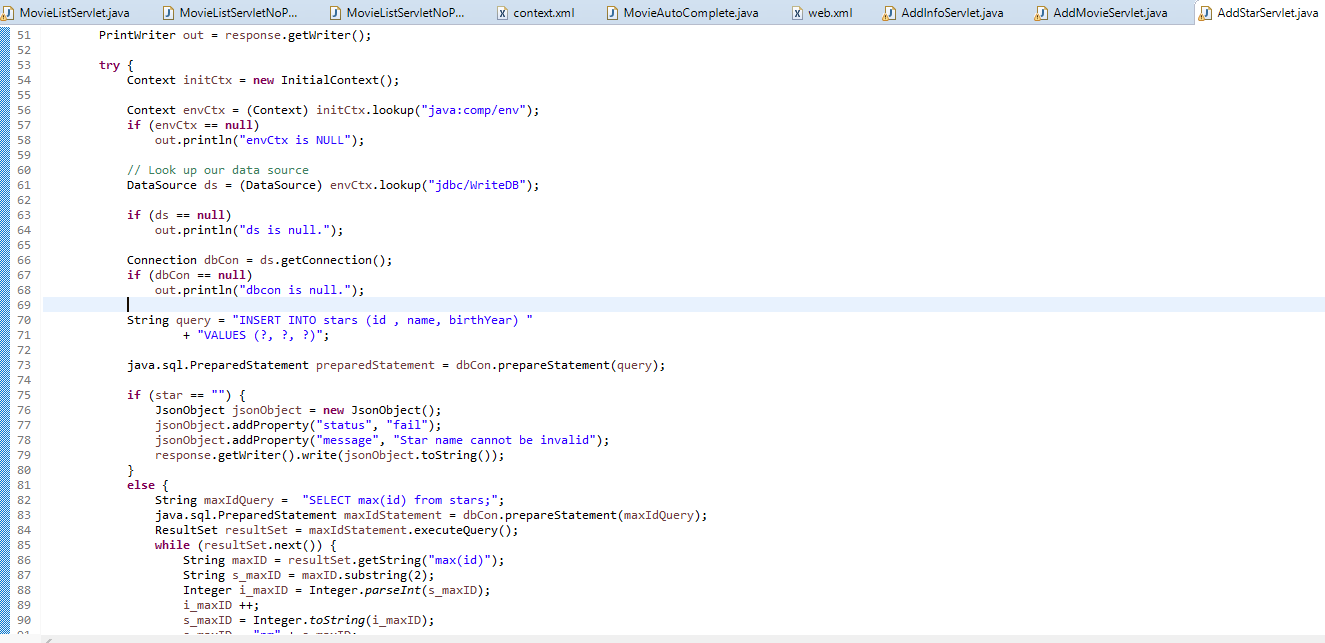
AddMovieServlet



AddInfoServlet



AddStarServlet



**Task 3**

* Have you uploaded the log files to Github? Where is it located

The log files are located in /project5/JMeter Testing/Logs

* Have you uploaded the HTML file (with all sections including analysis, written up) to Github? Where is it located?

The HTML file is located in /project5/JMeter Testing and in /project5/WebContent

* Have you uploaded the script  to Github? Where is it located?

The script is in /project5/JMeter Testing/Logs/8080 or /project5/JMeter Testing/Logs/80

* Have you uploaded the WAR file and README  to Github? Where is it located?

The WAR file and README are in the /project5