|  |  |  |
| --- | --- | --- |
| **Name:**  Sumukh Raju Bhat | **SRN:** PES1UG19CS519 |  |
|  |  |

**Problem Statement:**

#) The web-application mimics a simple calculator the 6 possible functionalities:

- Addition

- Subtraction

- Multiplication

- Division

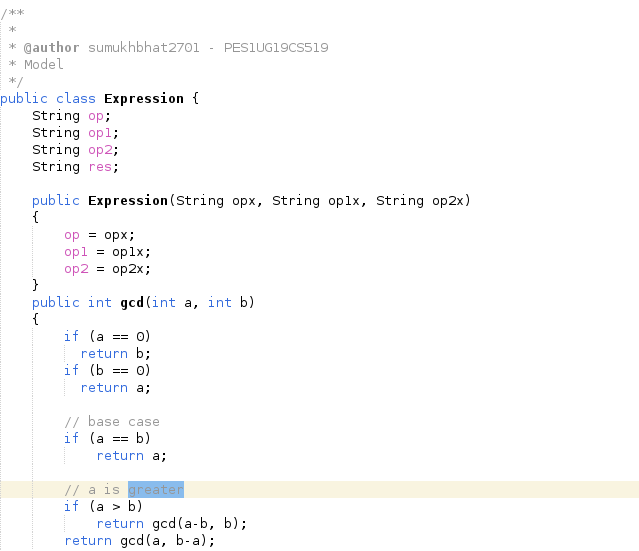
- Least Common Divisor

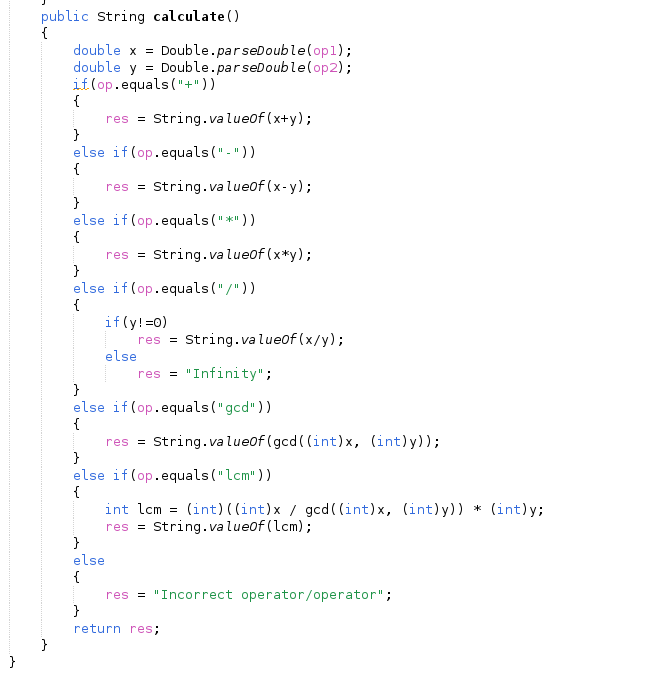
- Highest Common Factor

#) It takes in 2 operands and an operator to perform above functionalities.

#) Database to cache in the results to avoid re-calculation.

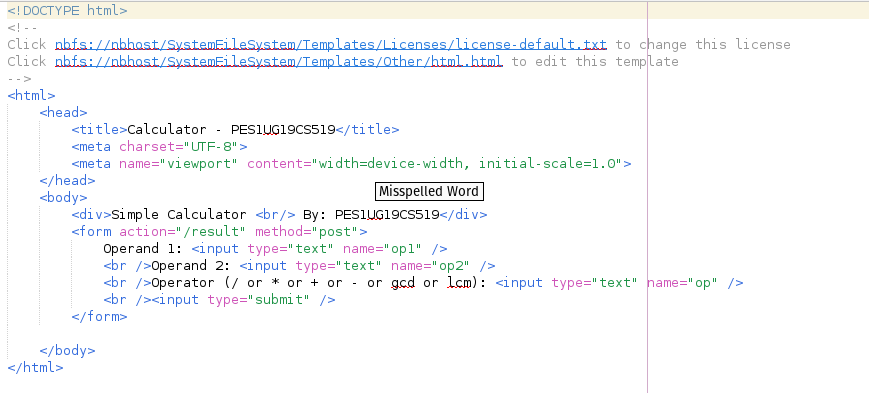
**Model Component:**

****

****

**View Component:**

****

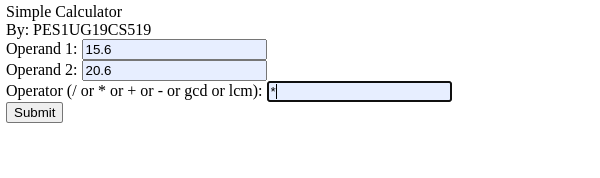
****

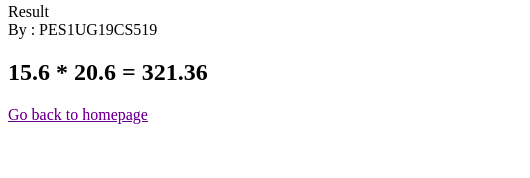
**Controller Component:**

****

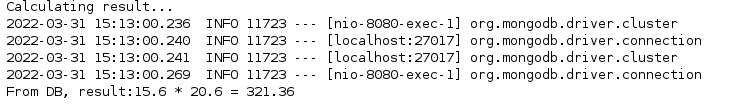
**Output Screen Shots:**

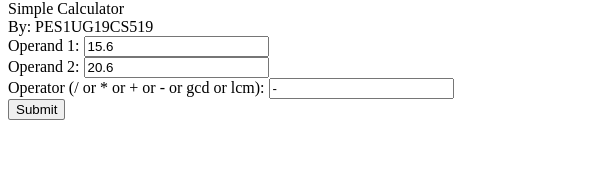
**1)**

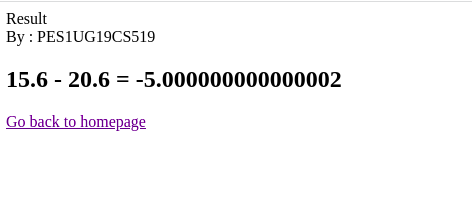
****

****

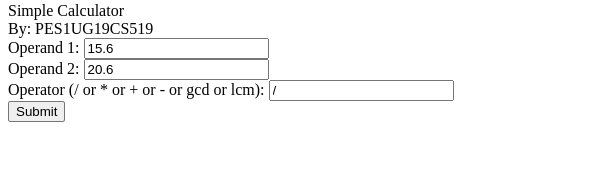
If the data is cached in the database, we get the following output in the terminal…

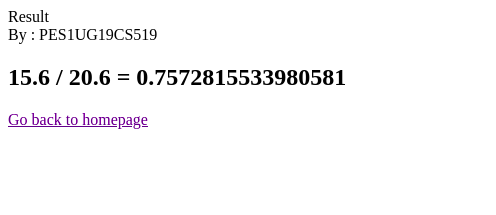
**2)**

****

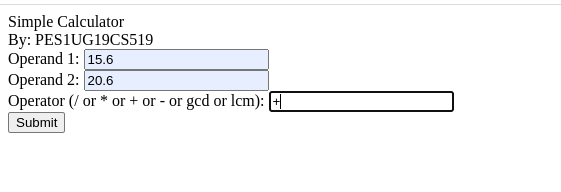
****

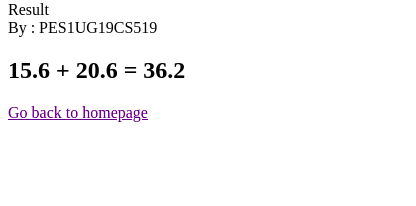
**3)**



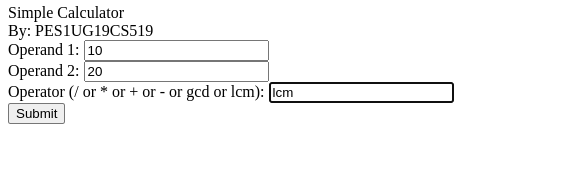


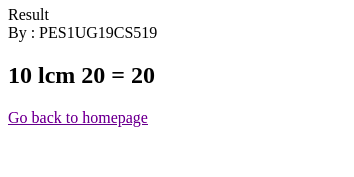
**4)**



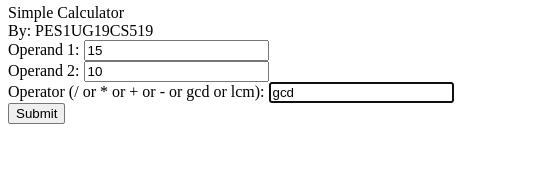


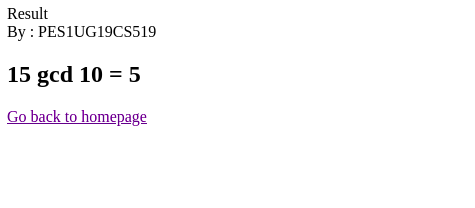
**5)**





**6)**





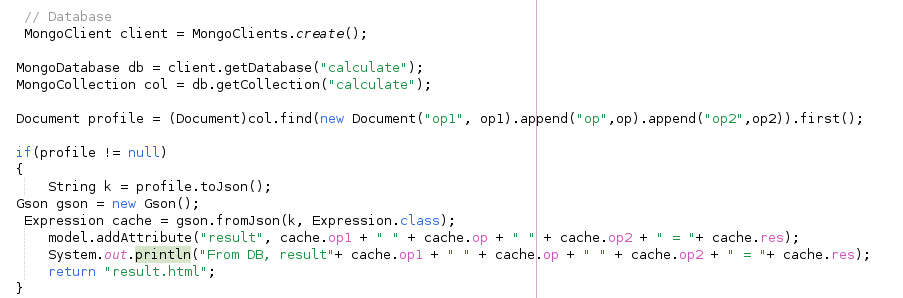
**Database:**

#) MongoDB, which is a document-based noSQL database is used because of its flexible schema.

#) MongoDB running on localhost is connected and a database called “calculate” is created/accessed and a collection also called as “calculate” is created/accessed.

#) Every time the calculation takes place, the results along with the operands and operators used for calculation. If in the future, if any expression is already cached, result is fetched from the database to avoid re-calculation.

#) Code:

****

**Technologies /Tools used:**

#) Spring framework is overall used to build this web-application.

#) Controller: Our custom Controller is built upon Spring’s org.springframework.stereotype.Controller. org.springframework.web.bind.annotation.PostMapping is used for handling post requests.

#) Model: Custom class called Expression is created and also spring’s org.springframework.ui.Model is used.

#) View: HTML pages serve as UI and thymeleaf is used to communicate with the backend.

#) Database: MongoDB as a database, which should be up and running in the localhost. MongoClient, MongoCollection, MongoDatabase are the modules to be imported to be connected to the database. Document module need to be imported as the result of database query is a document. Gson module to parse json objects got after parsing a document.

#) Server: Tomcat server running at port 8080 and localhost

#) IDE: Apache Netbeans v13.0 with Maven as a build-tool