Bonus Project

Installation and Execution

1. Download and unzip the folder ‘550_Bonus_G01393761’ to your preferred location.
2. Open your preferred Integrated Development Environment (IDE) such as IntelliJ, Eclipse, VsStudio, etc.
3. This is a Java Spring Boot project, ensure that you have Java Development Kit (JDK) installed and all path variables set up. If you need help with this, refer to the official documentation on **https://docs.oracle.com/cd/E19182-01/821-0917/inst_jdk_javahome_t/index.html**.
4. To start the application, locate and run the ‘Application.java’ file in the ‘src/main/java/com/bonusproject/’ directory using your IDE.
5. Once the application has successfully started, open your web browser and go to ‘**http://localhost:8080/**’. You can confirm that the application has started when the Run console prints: “com.bonusproject.Application : Started Application in { } seconds (JVM running for 2.486)”

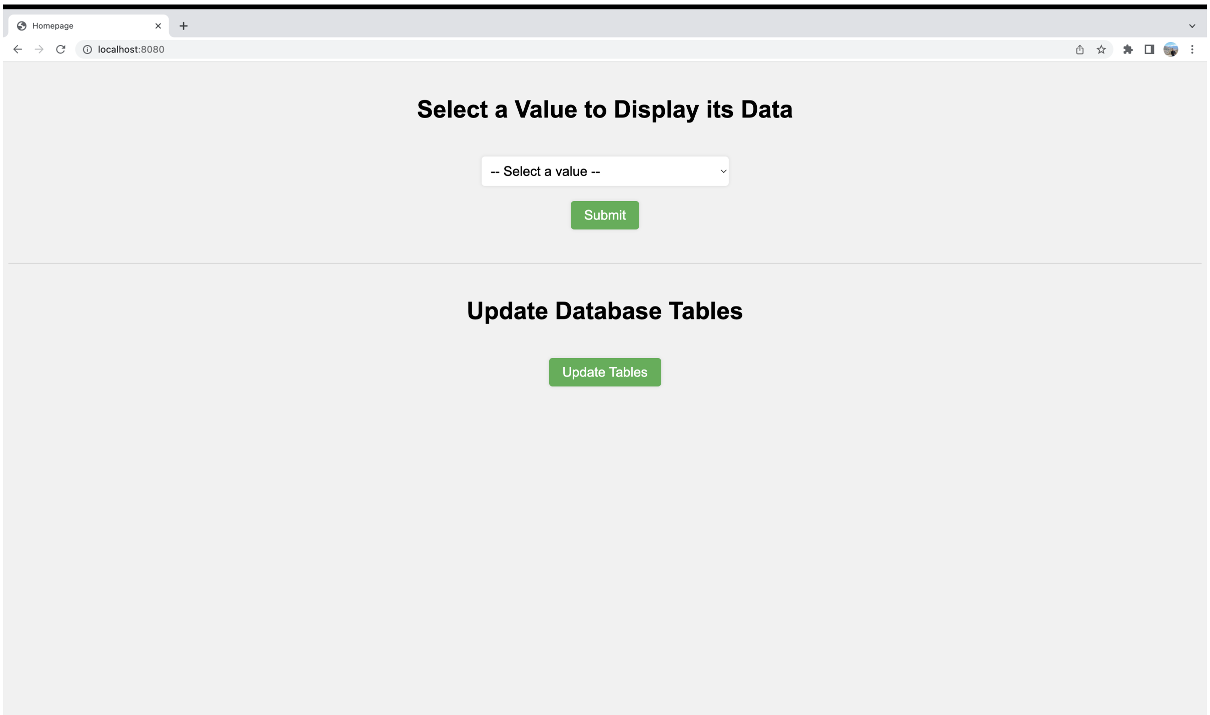
Background

- This is a software project written in Java using the Spring Boot framework. It allows users to interact with a pre-defined list of tables and views and execute Create, Read, Update, and Delete (CRUD) queries.
- The project uses H2, an in-memory database, to store and retrieve data. H2 is a lightweight and fast database management system that is popular for testing and prototyping purposes.
- When the application starts, it executes two files: `Schema.sql` and `Data.sql`. `Schema.sql` defines the structure of the database, including the tables and views, while `Data.sql` inserts data into the tables.

- The `ApiController.java` file handles requests from users to display data from a specific table or view. It searches the database and returns the result as a JSON object.
- The `UpdateTableController.java` file handles user-submitted queries and updates the database accordingly. It also displays the result of the query, whether it was successful or not, in the `result.html` file.
- The `index.html` file is the main page of the project. It provides links to the `displayall.html` and `update.html` pages.
- The `displayall.html` page displays the content of a table or view as a table. Users can select a table or view from a drop-down list and click the `Submit` button to view its content.
- The `update.html` page allows users to submit a custom query to update the database. Users can enter their query in the text box and click the `Execute` button to run it. The `result.html` page then displays the result of the query.
- Overall, this project provides a simple and user-friendly interface for interacting with a database using Java and the Spring Boot framework.

Working

- Upon successful startup of the application, the homepage can be accessed by visiting the URL '**http://localhost:8080/**'. The homepage comprises a menu that offers the option to view the contents of a pre-defined table or submit a query to be executed on the database.

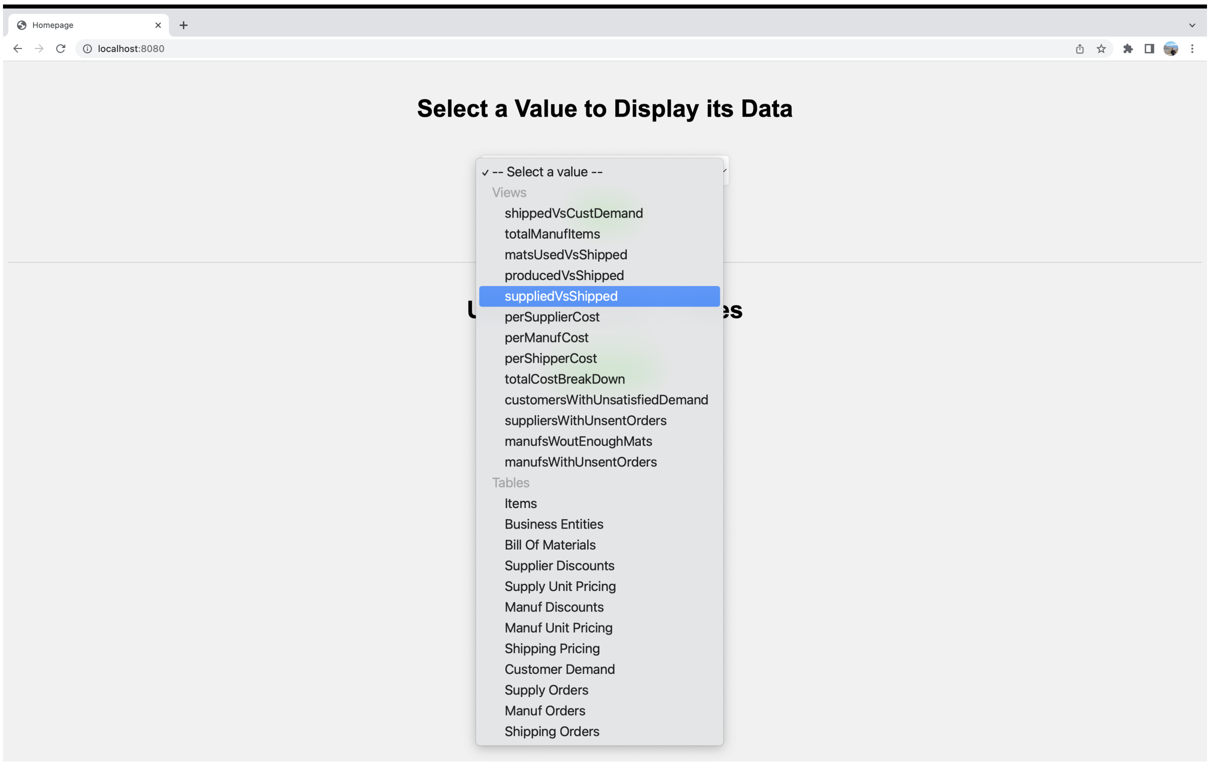


Select a Value to Display its Data

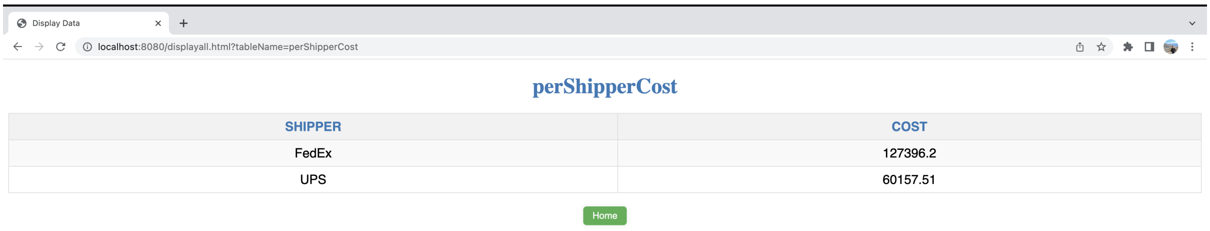
Update Database Tables

- To view the contents of a table, select the table from the dropdown list, and click the 'Submit' button. The table or view's contents are presented in a tabular format.

To return to the homepage, click the 'Home' button.

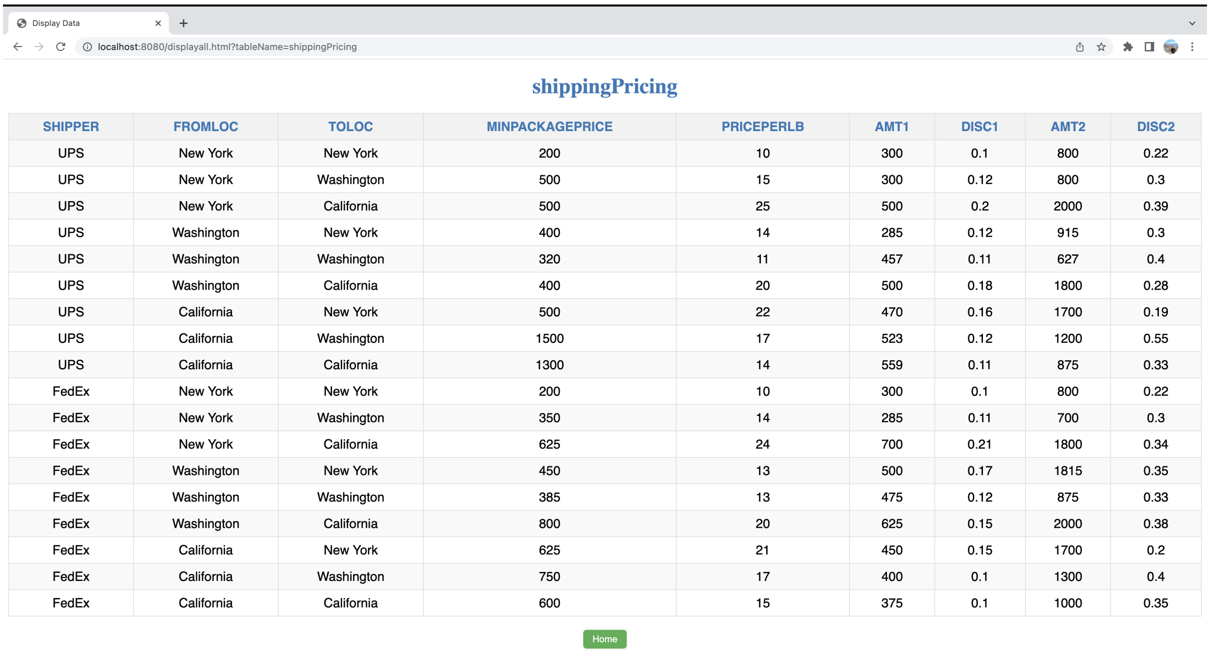


Select a Value to Display its Data



perShipperCost

SHIPPER	COST
FedEx	127396.2
UPS	60157.51

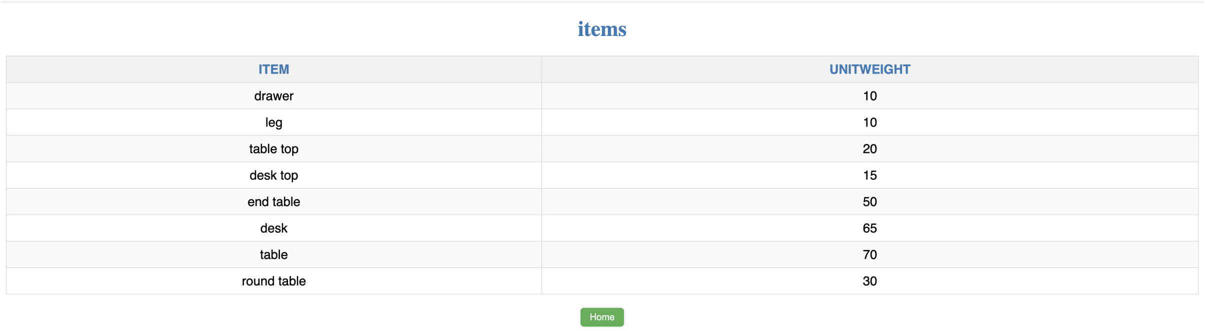


shippingPricing

SHIPPER	FROMLOC	TOLOC	MINPACKAGEPRICE	PRICEPERLB	AMT1	DISC1	AMT2	DISC2
UPS	New York	New York	200	10	300	0.1	800	0.22
UPS	New York	Washington	500	15	300	0.12	800	0.3
UPS	New York	California	500	25	500	0.2	2000	0.39
UPS	Washington	New York	400	14	285	0.12	915	0.3
UPS	Washington	Washington	320	11	457	0.11	627	0.4
UPS	Washington	California	400	20	500	0.18	1800	0.28
UPS	California	New York	500	22	470	0.16	1700	0.19
UPS	California	Washington	1500	17	523	0.12	1200	0.55
UPS	California	California	1300	14	559	0.11	875	0.33
FedEx	New York	New York	200	10	300	0.1	800	0.22
FedEx	New York	Washington	350	14	285	0.11	700	0.3
FedEx	New York	California	625	24	700	0.21	1800	0.34
FedEx	Washington	New York	450	13	500	0.17	1815	0.35
FedEx	Washington	Washington	385	13	475	0.12	875	0.33
FedEx	Washington	California	800	20	625	0.15	2000	0.38
FedEx	California	New York	625	21	450	0.15	1700	0.2
FedEx	California	Washington	750	17	400	0.1	1300	0.4
FedEx	California	California	600	15	375	0.1	1000	0.35

- Selecting the 'Update Tables' button redirects you to a page where you can submit a query to be executed on the database. After entering your query, click on the 'Execute' button to run the query on the database.
- Example:

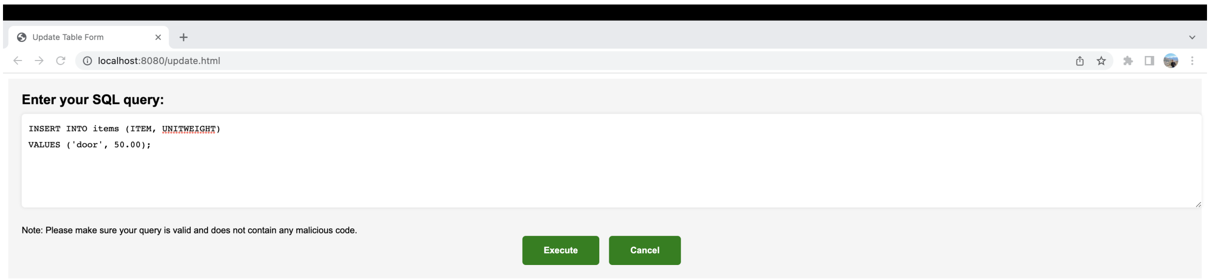
a. Initial ‘Items’ table



items

ITEM	UNITWEIGHT
drawer	10
leg	10
table top	20
desk top	15
end table	50
desk	65
table	70
round table	30

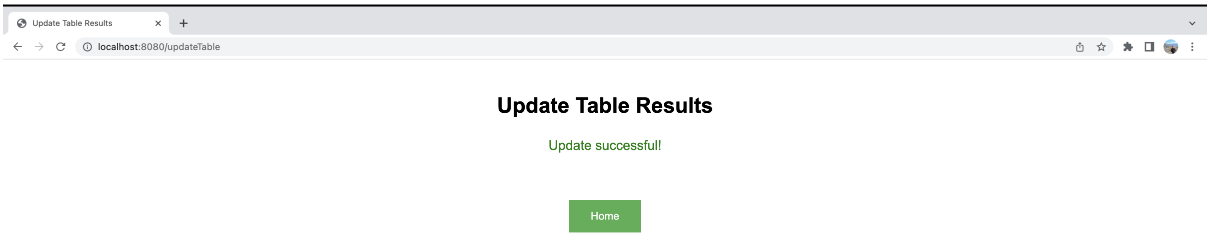
b. Running insert query



Enter your SQL query:

```
INSERT INTO items (ITEM, UNITWEIGHT)
VALUES ('door', 50.00);
```

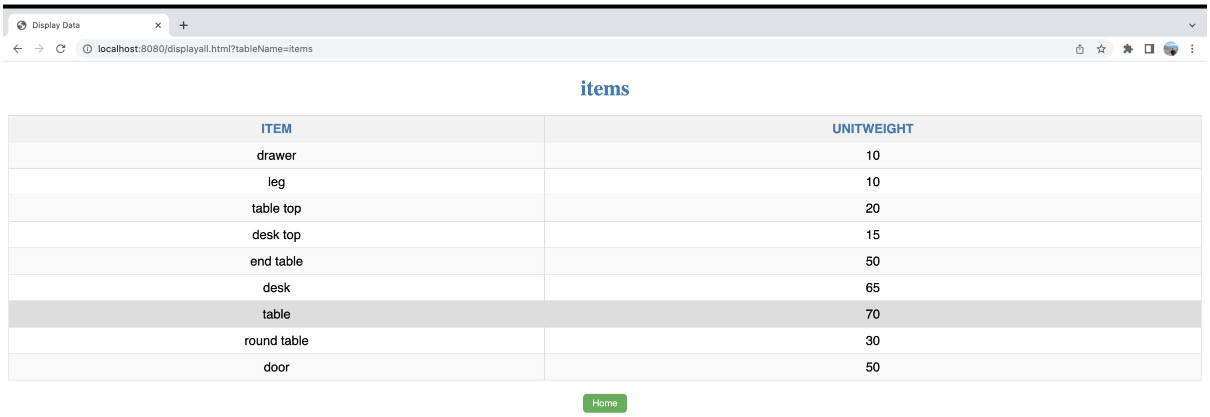
Note: Please make sure your query is valid and does not contain any malicious code.



Update Table Results

Update successful!

c. For instance, if you insert data into the 'Items' table using a query, you can view the added data by accessing the 'Items' table. This can also be achieved by directly visiting the URL **http://localhost:8080/displayall.html?tableName=items**).

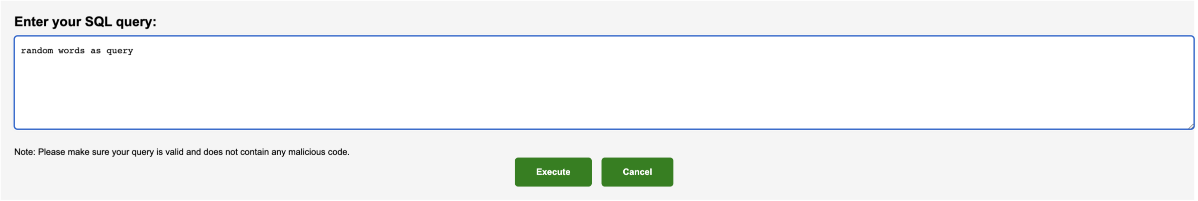


items

ITEM	UNITWEIGHT
drawer	10
leg	10
table top	20
desk top	15
end table	50
desk	65
table	70
round table	30
door	50

Here record 'door' was inserted.

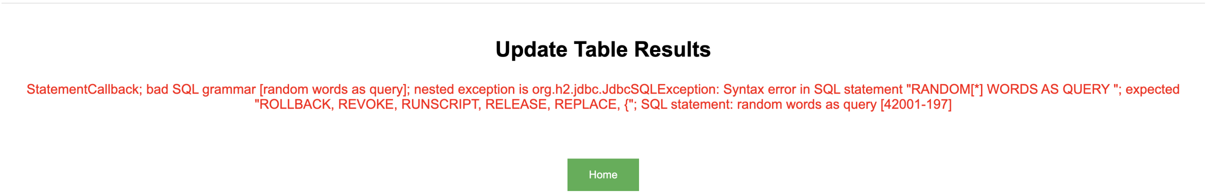
d. Submitting an incorrect query results in an error message indicating the issue with the query.



Enter your SQL query:

```
random words as query
```

Note: Please make sure your query is valid and does not contain any malicious code.



Update Table Results

StatementCallback; bad SQL grammar [random words as query]; nested exception is org.h2.jdbc.JdbcSQLException: Syntax error in SQL statement "RANDOM[*] WORDS AS QUERY "; expected "ROLLBACK, REVOKE, RUNSCRIPT, RELEASE, REPLACE, {"; SQL statement: random words as query [42001-197]