

CS313: Lab Assignment 6

Part B

B Siddharth Prabhu

200010003@iitdh.ac.in

11 October 2022

1 Setup and Installation

- Download the project `J2EEProject.zip` file from the Part B folder of `200010003.zip`. Extract to a specific directory.
- Launch Eclipse. Ensure that no other project open in the workspace has the same name as this project (J2EEProject). Then, click on File → Open Project From File System.
- Choose the directory of the extracted project in the import source section and click on Finish. Ensure that the selected folder contains the directories such as `src` and `.settings`, and isn't further nested.
- Errors, if any, can be fixed using the instructions given in the handout.
- Right click the project name, then click Build Path → Configure build path. Click on Project Facets on the left panel, then hit the Runtime section on the right section of the window and choose apache tomcat server and click on the Apply and Close button.
- To run the application, right click the project → Run as → Run on server. Once the server is up, access the following URL: `http://localhost:8080/J2EEProject/Home.jsp`
- The workspace should look something like this:

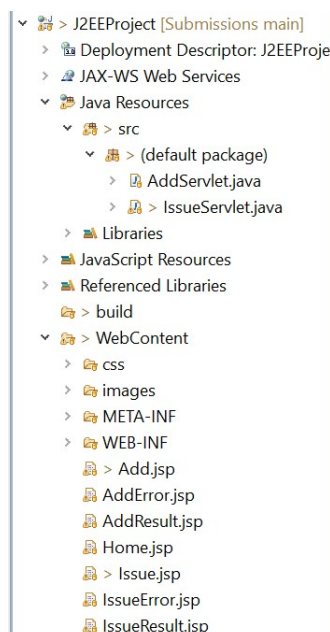


Figure 1: Image of Workspace

2 Design of the Library schema

Given database schema is as follows:

```
book (book_id, title, category, author);
student (student_id, name, dept_name, year, semester);
issue (student_id, book_id, issue_date, return_date);
```

Firstly, we create a user 'librarian' using the code in `b0.sql`, which is as follows:

```
CREATE USER 'librarian'@'localhost' IDENTIFIED BY 'eselifterbraun';
CREATE DATABASE library;
GRANT ALL PRIVILEGES ON library.* TO 'librarian'@'localhost';
```

Output of the same, followed by logging in through the new user:

```
mysql> CREATE USER 'librarian'@'localhost' IDENTIFIED BY 'eselifterbraun';
Query OK, 0 rows affected (0.42 sec)

mysql> CREATE DATABASE library;
Query OK, 1 row affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON library.* TO 'librarian'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Figure 2: Creation of user 'librarian'

```
siddharth@DESKTOP-5490SID:/mnt/c/Users/bsidd/Desktop/CS313 DBIS_Lab/Submissions/Assignment-6 (Project-1)$ mysql -u lib
rian -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 110
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| library |
| performance_schema |
+-----+
3 rows in set (0.01 sec)

mysql> |
```

Figure 3: Logging in to new user

Now, we run the script `b1.sql` to create the tables. This script contains the following:

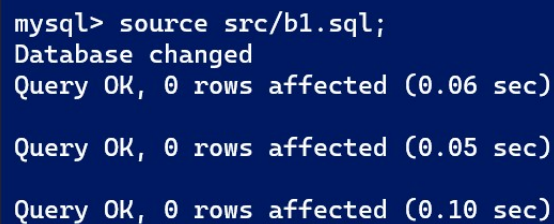
```
USE library;

CREATE TABLE book(
    book_id VARCHAR(6) PRIMARY KEY,
    title VARCHAR(50) NOT NULL,
    category VARCHAR(20),
    author VARCHAR(50)
);

CREATE TABLE student(
    student_id VARCHAR(5) PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    dept_name VARCHAR(20),
    year YEAR,
    semester NUMERIC(2)
);

CREATE TABLE issue(
    student_id VARCHAR(5),
    book_id VARCHAR(6),
    issue_date DATE NOT NULL,
    return_date DATE,
    PRIMARY KEY(student_id, book_id, issue_date),
    FOREIGN KEY(student_id) REFERENCES student(student_id),
    FOREIGN KEY(book_id) REFERENCES book(book_id)
);
```

After running this script using `source b1.sql`, where the file `b1.sql` is in the current working directory, the output is as follows:



```
mysql> source src/b1.sql;
Database changed
Query OK, 0 rows affected (0.06 sec)

Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.10 sec)
```

Figure 4: Creation of tables using `b1.sql`

3 Loading values into the database tables

Now, we run the script `b2.sql` to load data into the tables. It contains the following queries:

```
INSERT INTO book VALUES (
    "200001", "Hunger Games", "Sci-Fi", "Suzanne Collins");
INSERT INTO book VALUES (
    "200002", "Percy Jackson", "Adventure", "Rick Riordan");
INSERT INTO book VALUES (
    "200003", "Harry Potter", "Fantasy", "JK Rowling");
INSERT INTO book VALUES (
    "200004", "Prisoners of Geography", "Non-fiction", "Tim Marshall");
INSERT INTO book VALUES (
    "200005", "Origin Story", "Non-fiction", "David Christian");
INSERT INTO book VALUES (
    "200006", "The Psychology of Persuasion", "Non-fiction", "Robert B Cialdini");
INSERT INTO book VALUES (
    "200007", "To Kill a Mockingbird", "Classic", "Harper Lee");

INSERT INTO student VALUES ("10002", "Altmash", "CSE", 2020, 5);
INSERT INTO student VALUES ("10004", "Sourabh", "CSE", 2020, 5);
INSERT INTO student VALUES ("10012", "Devdatt", "CSE", 2020, 5);
INSERT INTO student VALUES ("20003", "Aditya", "EE", 2020, 5);
INSERT INTO student VALUES ("30003", "Abhishek", "MMAE", 2020, 5);
INSERT INTO student VALUES ("11003", "Akash", "CSE", 2021, 3);
INSERT INTO student VALUES ("21003", "Anand", "EE", 2021, 3);
INSERT INTO student VALUES ("21008", "Amit", "EE", 2021, 3);
INSERT INTO student VALUES ("22003", "Harsh", "EE", 2022, 1);
INSERT INTO student VALUES ("32003", "Manjunath", "MMAE", 2022, 1);
INSERT INTO student VALUES ("22006", "Nagesh", "EE", 2022, 1);

INSERT INTO issue VALUES ("10012", "200001", "2022-09-01", "2022-09-15");
INSERT INTO issue VALUES ("11003", "200001", "2022-10-10", "2022-10-25");
INSERT INTO issue VALUES ("21003", "200003", "2022-09-01", "2022-09-15");
INSERT INTO issue VALUES ("30003", "200004", "2022-09-01", "2022-09-15");
INSERT INTO issue VALUES ("10012", "200005", "2022-09-15", "2022-09-29");
INSERT INTO issue VALUES ("32003", "200006", "2022-09-01", "2022-09-15");
INSERT INTO issue VALUES ("21008", "200006", "2022-08-01", "2022-08-15");
INSERT INTO issue VALUES ("20003", "200006", "2022-09-01", "2022-09-15");
INSERT INTO issue VALUES ("22003", "200001", "2022-08-01", "2022-08-15");
INSERT INTO issue VALUES ("10002", "200007", "2022-09-01", "2022-09-15");
INSERT INTO issue VALUES ("30003", "200007", "2022-09-16", "2022-09-30");
```

After running this script using `source b2.sql`, where the file `b2.sql` is in the current working directory, we obtain the output in Figure (5).

```
mysql> source src/b2.sql;
Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)
```

Figure 5: Loading data into tables using `b2.sql`

After the loading, the tables are populated as follows:

```
mysql> select * from book;
+-----+-----+-----+-----+
| book_id | title                | category | author          |
+-----+-----+-----+-----+
| 200001 | Hunger Games         | Sci-Fi   | Suzanne Collins |
| 200002 | Percy Jackson        | Adventure | Rick Riordan    |
| 200003 | Harry Potter         | Fantasy  | JK Rowling       |
| 200004 | Prisoners of Geography | Non-fiction | Tim Marshall    |
| 200005 | Origin Story         | Non-fiction | David Christian  |
| 200006 | The Psychology of Persuasion | Non-fiction | Robert B Cialdini |
| 200007 | To Kill a Mockingbird | Classic  | Harper Lee       |
+-----+-----+-----+-----+
7 rows in set (0.04 sec)
```

Figure 6: Contents of table 'book'

```
mysql> select * from student;
+-----+-----+-----+-----+-----+
| student_id | name      | dept_name | year | semester |
+-----+-----+-----+-----+-----+
| 10002      | Altmash   | CSE       | 2020 | 5         |
| 10004      | Sourabh   | CSE       | 2020 | 5         |
| 10012      | Devdatt   | CSE       | 2020 | 5         |
| 11003      | Akash     | CSE       | 2021 | 3         |
| 20003      | Aditya    | EE        | 2020 | 5         |
| 21003      | Anand     | EE        | 2021 | 3         |
| 21008      | Amit      | EE        | 2021 | 3         |
| 22003      | Harsh     | EE        | 2022 | 1         |
| 22006      | Nagesh    | EE        | 2022 | 1         |
| 30003      | Abhishek  | MMAE      | 2020 | 5         |
| 32003      | Manjunath | MMAE      | 2022 | 1         |
+-----+-----+-----+-----+-----+
11 rows in set (0.01 sec)
```

Contents of table 'student'

```
mysql> select * from issue;
+-----+-----+-----+-----+
| student_id | book_id | issue_date | return_date |
+-----+-----+-----+-----+
| 10002      | 200007 | 2022-09-01 | 2022-09-15 |
| 10012      | 200001 | 2022-09-01 | 2022-09-15 |
| 10012      | 200005 | 2022-09-15 | 2022-09-29 |
| 11003      | 200001 | 2022-09-21 | 2022-10-05 |
| 11003      | 200001 | 2022-10-10 | 2022-10-24 |
| 20003      | 200006 | 2022-09-01 | 2022-09-15 |
| 21003      | 200003 | 2022-09-01 | 2022-09-15 |
| 21008      | 200006 | 2022-08-01 | 2022-08-15 |
| 22003      | 200001 | 2022-08-01 | 2022-08-15 |
| 30003      | 200004 | 2022-09-01 | 2022-09-15 |
| 30003      | 200007 | 2022-09-16 | 2022-09-30 |
| 32003      | 200006 | 2022-09-01 | 2022-09-15 |
+-----+-----+-----+-----+
12 rows in set (0.01 sec)
```

Contents of table 'issue'

Figure 7: Contents of other tables

4 Creation of a Java J2EE Project

Firstly, we will take a look at the 3 main web pages associated with our Library Management System. Then, we will interact with the web pages by filling forms, and view the results obtained. Lastly, we will note the measures taken to ensure constraint checking.

4.1 A view of the web pages created

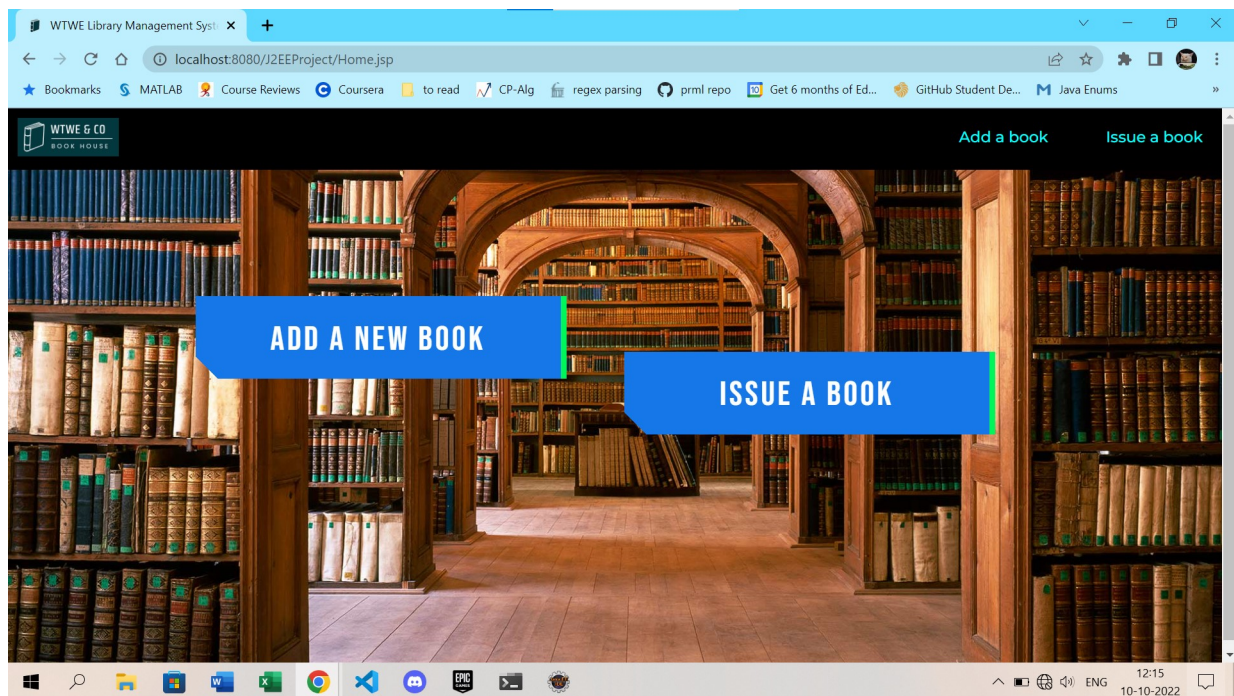


Figure 8: Home.jsp

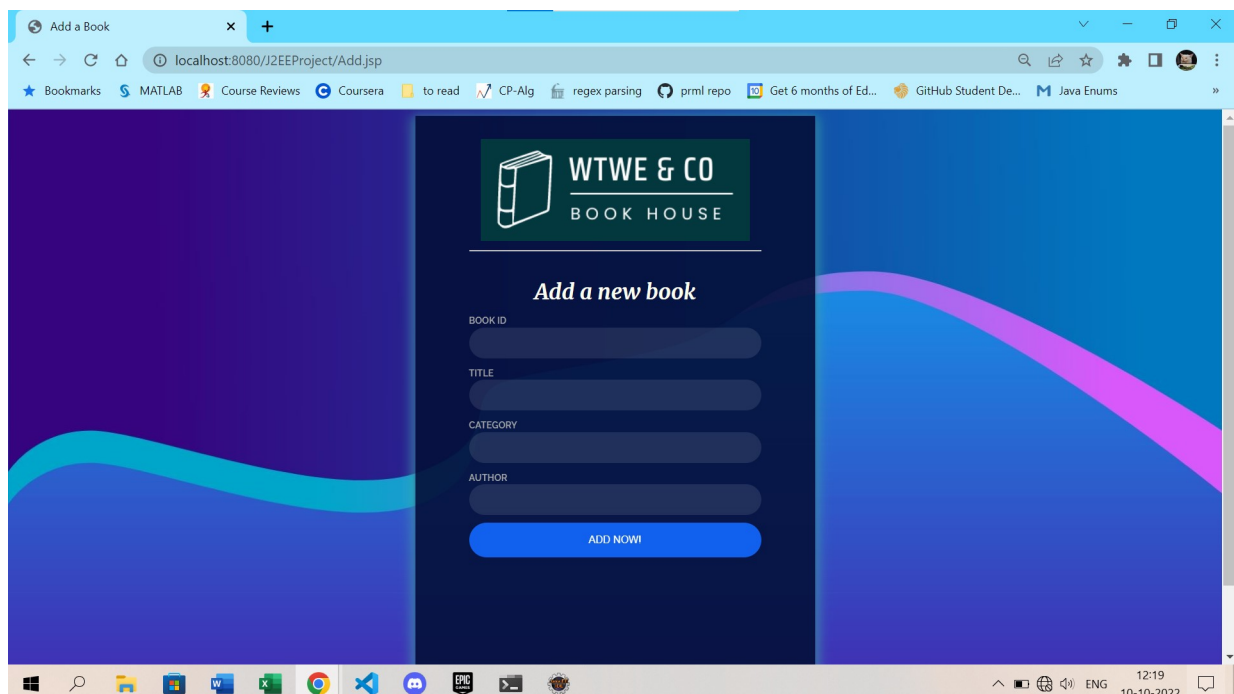


Figure 9: Add.jsp

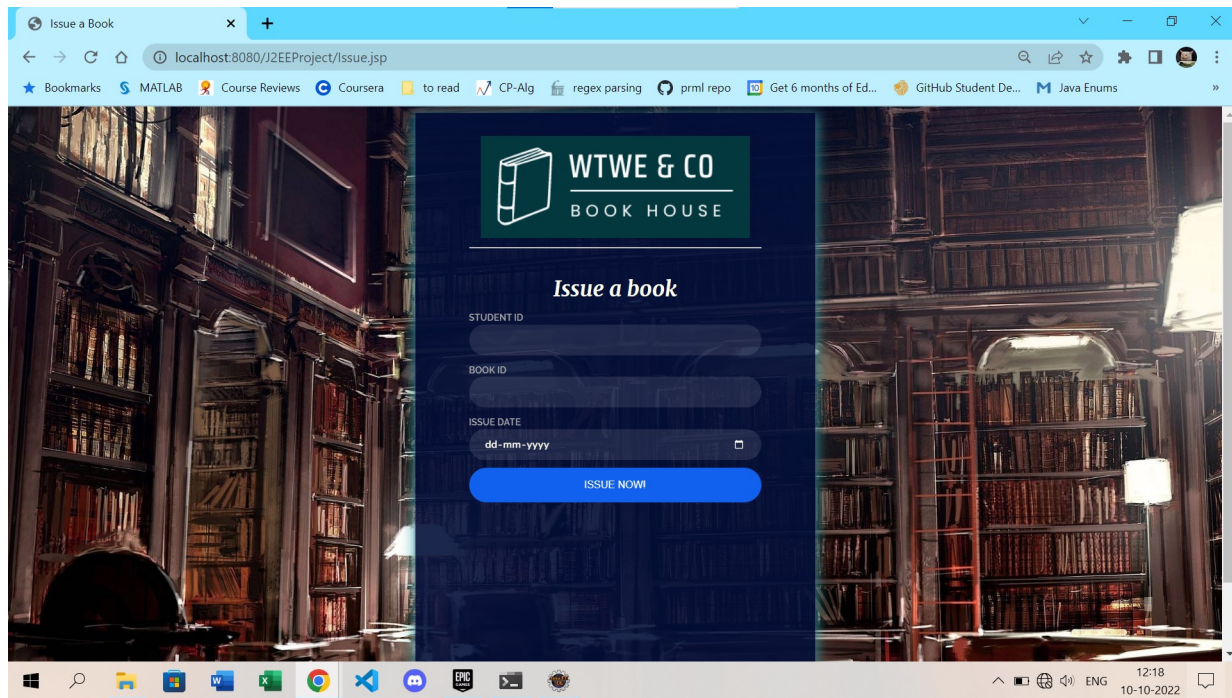


Figure 10: Issue .jsp

4.2 Adding a book to the database using 'Add.jsp'

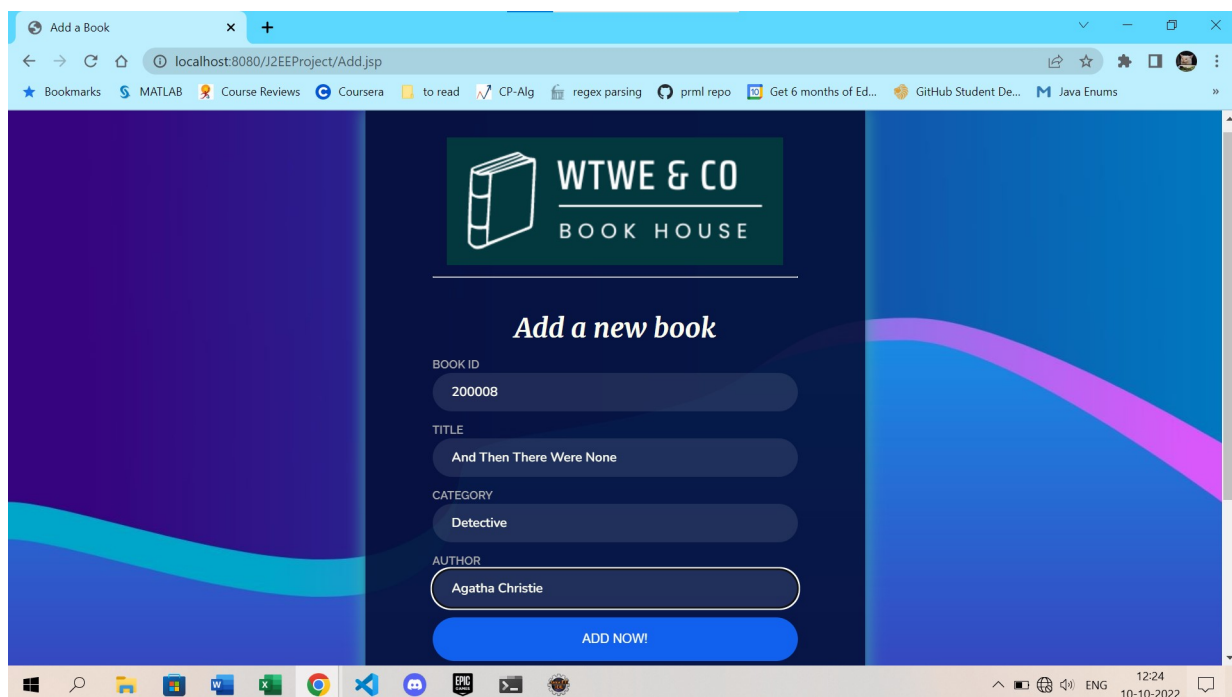


Figure 11: Input to the 'Add' form

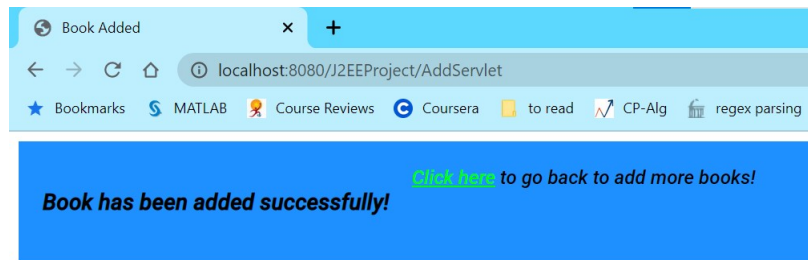


Figure 12: Added book successfully!

```
mysql> select * from book;
```

book_id	title	category	author
200001	Hunger Games	Sci-Fi	Suzanne Collins
200002	Percy Jackson	Adventure	Rick Riordan
200003	Harry Potter	Fantasy	JK Rowling
200004	Prisoners of Geography	Non-fiction	Tim Marshall
200005	Origin Story	Non-fiction	David Christian
200006	The Psychology of Persuasion	Non-fiction	Robert B Cialdini
200007	To Kill a Mockingbird	Classic	Harper Lee
200008	And Then There Were None	Detective	Agatha Christie

8 rows in set (0.01 sec)

Figure 13: Updated 'book' table

4.3 Issuing a book using 'Issue.jsp'

Issue a Book

localhost:8080/J2EEProject/Issue.jsp

Bookmarks: MATLAB, Course Reviews, Coursera, to read, CP-Alg, regex parsing, prml repo, Get 6 months of Ed..., GitHub Student De..., Java Enums

WTWE & CO BOOK HOUSE

Issue a book

STUDENT ID: 11003

BOOK ID: 200008

ISSUE DATE: 01-10-2022

ISSUE NOW!

Windows taskbar: 12:29, 10-10-2022

Figure 14: Input to the 'Issue' form

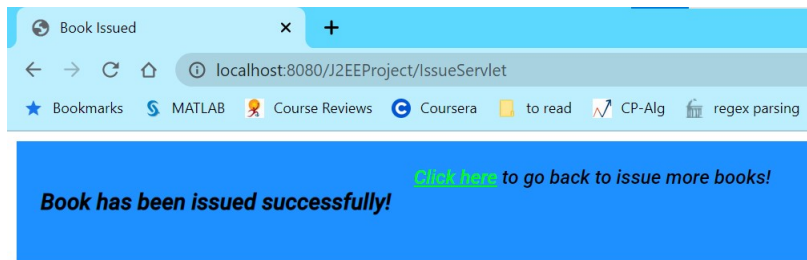


Figure 15: Issued book successfully!

```
mysql> select * from issue;
```

student_id	book_id	issue_date	return_date
10002	200007	2022-09-01	2022-09-15
10012	200001	2022-09-01	2022-09-15
10012	200005	2022-09-15	2022-09-29
11003	200001	2022-09-21	2022-10-05
11003	200001	2022-10-10	2022-10-24
11003	200008	2022-10-01	2022-10-15
20003	200006	2022-09-01	2022-09-15
21003	200003	2022-09-01	2022-09-15
21008	200006	2022-08-01	2022-08-15
22003	200001	2022-08-01	2022-08-15
30003	200004	2022-09-01	2022-09-15
30003	200007	2022-09-16	2022-09-30
32003	200006	2022-09-01	2022-09-15

13 rows in set (0.00 sec)

Figure 16: Updated 'issue' table

4.4 Failed add due to pre-existing book_id

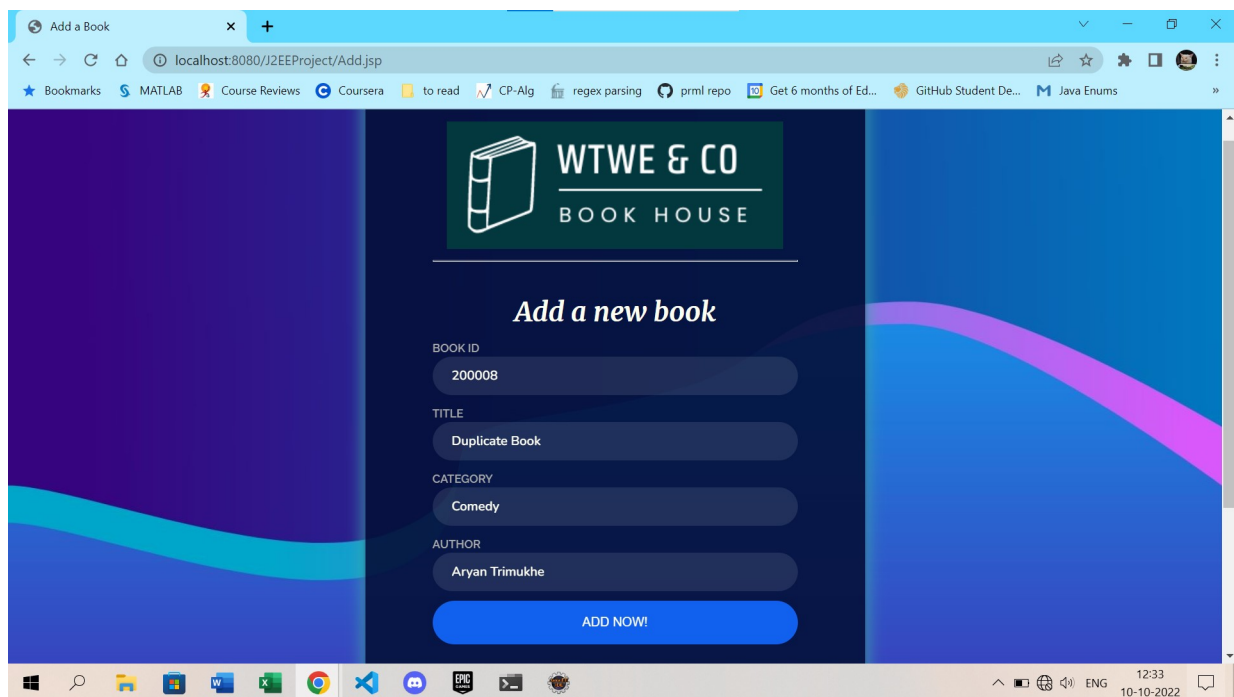


Figure 17: Input with pre-existing book_id

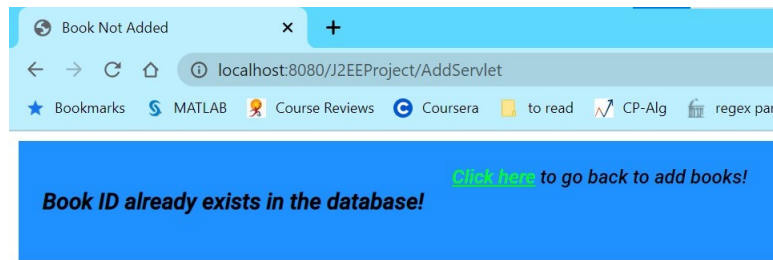


Figure 18: Could not add book!

4.5 Failed issue due to nonexistent `book_id` or `student_id`

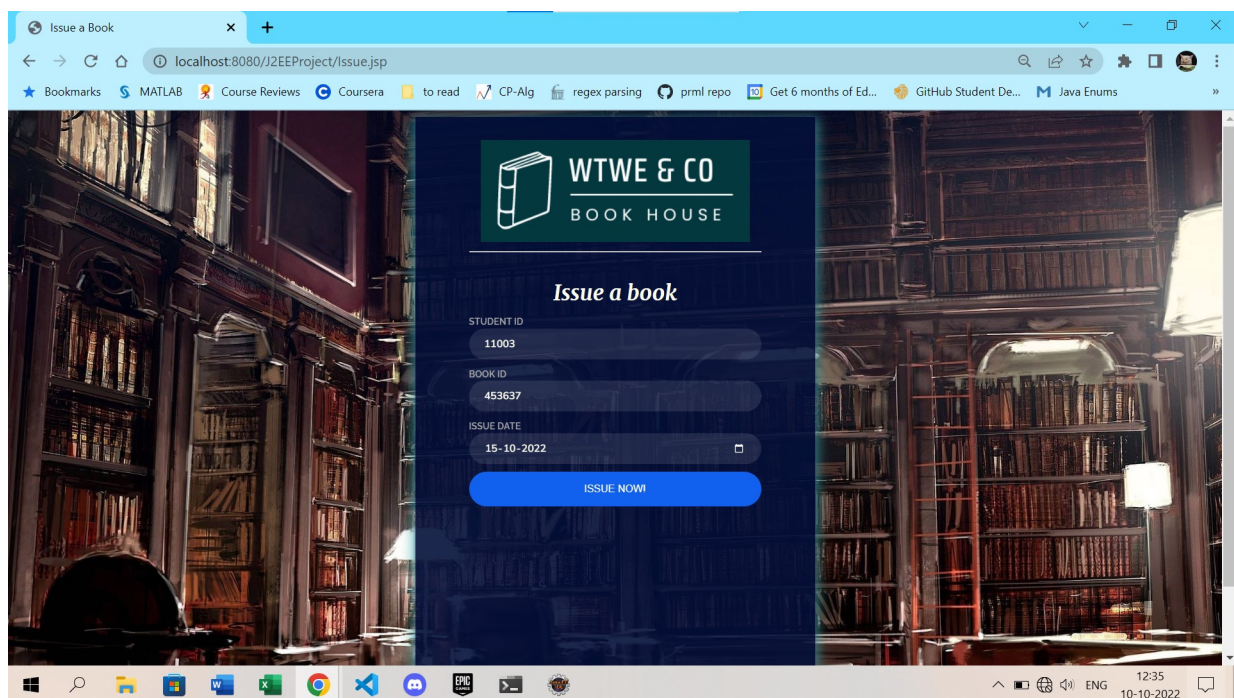


Figure 19: Input with nonexistent `book_id`

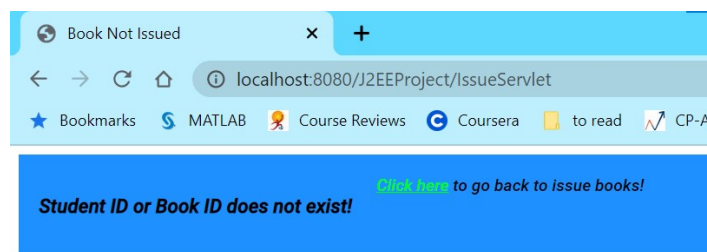


Figure 20: Could not issue book!

5 Remarks about integrity constraints and default values

- Initially, return date is set to `null` when issuing a book. Then, an update statement is called to set return date of the new issue record to 14 days ahead of the issue date. The number of days could also be taken as input from the form, but for our purposes this is enough.
- The jsp form requires that all fields be non-empty for submission to be possible, by setting input as "required".
- During adding of book, Java program checks if book with given id already exists in the database. It tells the user the outcome accordingly.
- During issuing of book, Java program checks if student with given id (and also book with given id) exists in the database. Then, it tells the user the outcome accordingly.
- Referential integrity constraint violations would not occur since we do checking of records according to the above two points.