

SWE425-eRegistrar-Presentation

1. Database table structure

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Schemas):** Shows the database structure under the schema 'cs-cs425-eregistrar'. The 'courses' table is selected.
- Central Panel (Result Grid):** Displays the results of the query: `SELECT * FROM `cs-cs425-eregistrar`.courses;`. The data is as follows:

course_id	course_code	course_title	pass_grade_point	retakeable_grade_point
1	CS390	Fundamental Programming Practices	3	2.7
2	CS401	Modern Programming Practices	3	2.7
3	CS523	Big Data Technology	2.3	2.3
4	CS435	Algorithms	2.7	2.3

- Bottom Panel (Object Info):** Provides detailed information about the 'courses' table, including columns: course_id (bigint AI PK), course_code (varchar(255)), course_title (varchar(255)), pass_grade_point (double), and retakeable_grade_point (double).
- Right Panel (Help):** A note states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help".

2. eRegistrar Restful API (create academic block)

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections and environments. In the main workspace, a POST request is being prepared to 'http://localhost:8081/eregistrar/api/academic-block-management/create'. The 'Body' tab is selected, displaying the following JSON payload:

```
1 {
2   "blockCode": "Dec-2021",
3   "blockTitle": "December 2021",
4   "startDate": "2021-12-01",
5   "endDate": "2021-12-31"
6 }
```

Below the body, there's a response placeholder with a cartoon character pointing at it and the text 'Click Send to get a response'.

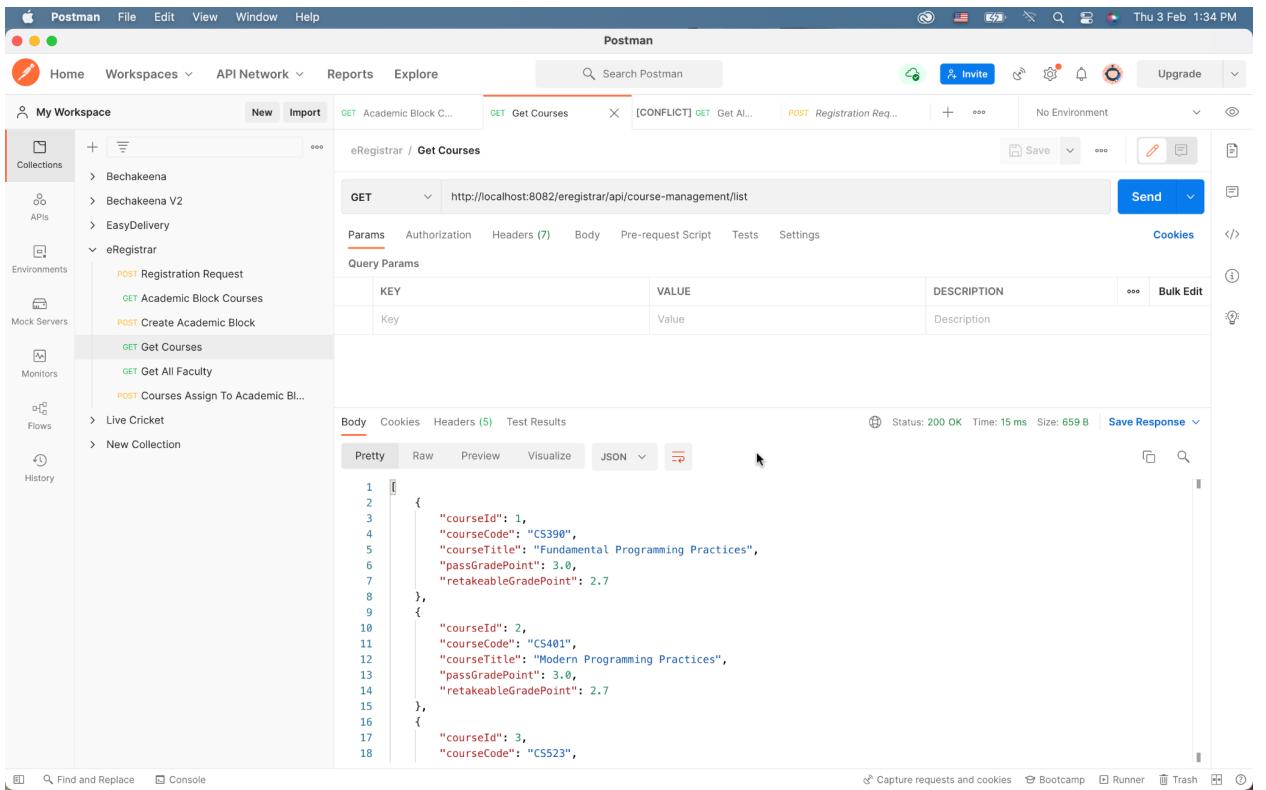
3. eRegistrar Restful API (academic block list)

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections and environments. In the main workspace, a GET request is being prepared to 'http://localhost:8082/eregistrar/api/academic-block-management/list'. The 'Body' tab is selected, showing the following JSON response:

```
1 {
2   "blockId": 1,
3   "blockCode": "Sep-2021",
4   "blockTitle": "September 2021",
5   "startDate": "2021-09-01",
6   "endDate": "2021-09-30",
7   "blockCourses": [
8     {
9       "blockCourseId": 1,
10      "capacity": 40,
11      "user": [
12        {
13          "userId": 1,
14          "userName": "613473",
15          "firstName": "Md Anwar",
16          "lastName": "Hossain"
17        }
18      ],
19      "course": {
20        "courseId": 1,
21        "courseName": "Computer Science"
22      }
23    }
24  ]
25 }
```

The status bar at the bottom indicates a successful response: Status: 200 OK, Time: 274 ms, Size: 1.05 KB.

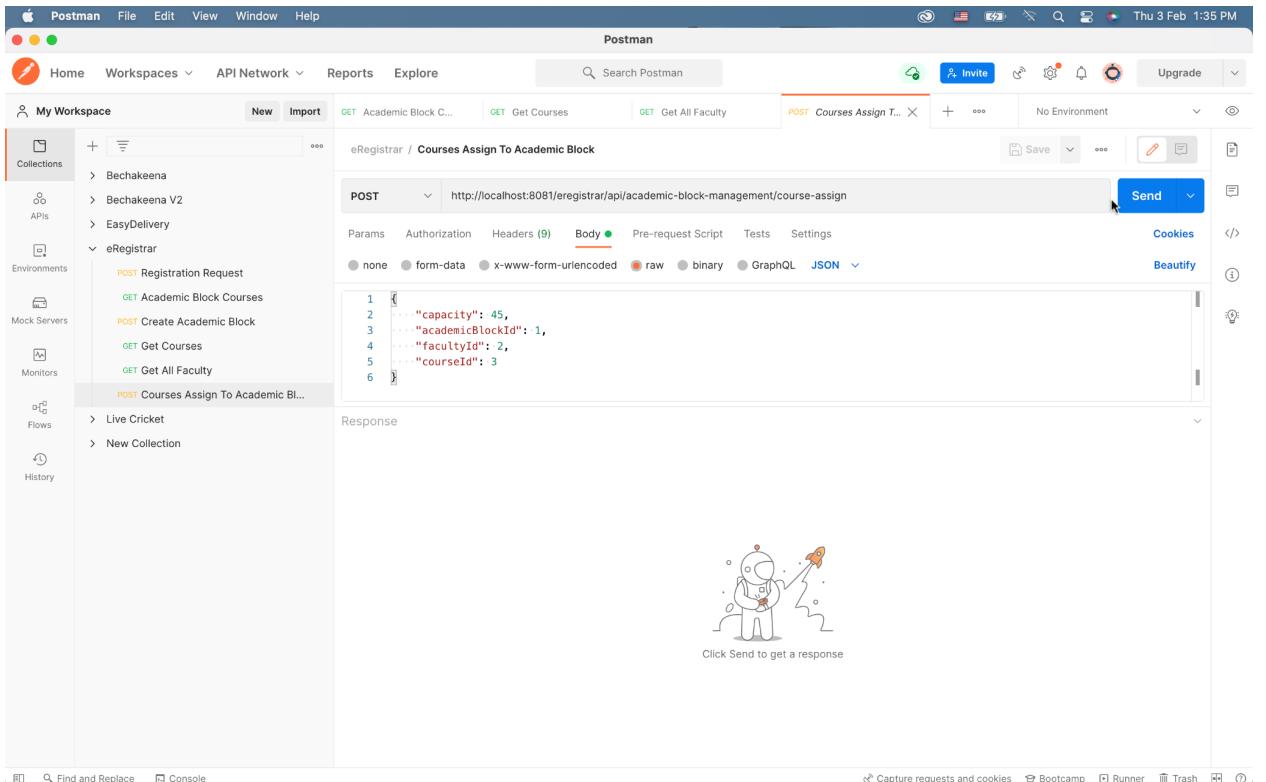
4. eRegistrar Restful API (course list)



The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists several collections, including 'eRegistrar' which contains 'Get Courses', 'Get All Faculty', and 'POST Courses Assign To Academic Bl...'. The main workspace displays a 'GET / Get Courses' request. The URL is set to `http://localhost:8082/eregistrar/api/course-management/list`. The 'Body' tab shows a JSON response with three course entries:

```
1  [
2    {
3      "courseId": 1,
4      "courseCode": "CS398",
5      "courseTitle": "Fundamental Programming Practices",
6      "passGradePoint": 3.0,
7      "retakeableGradePoint": 2.7
8    },
9    {
10      "courseId": 2,
11      "courseCode": "CS401",
12      "courseTitle": "Modern Programming Practices",
13      "passGradePoint": 3.0,
14      "retakeableGradePoint": 2.7
15    },
16    {
17      "courseId": 3,
18      "courseCode": "CS523",
```

5. eRegistrar Restful API (course assign to academic block also assign faculty)

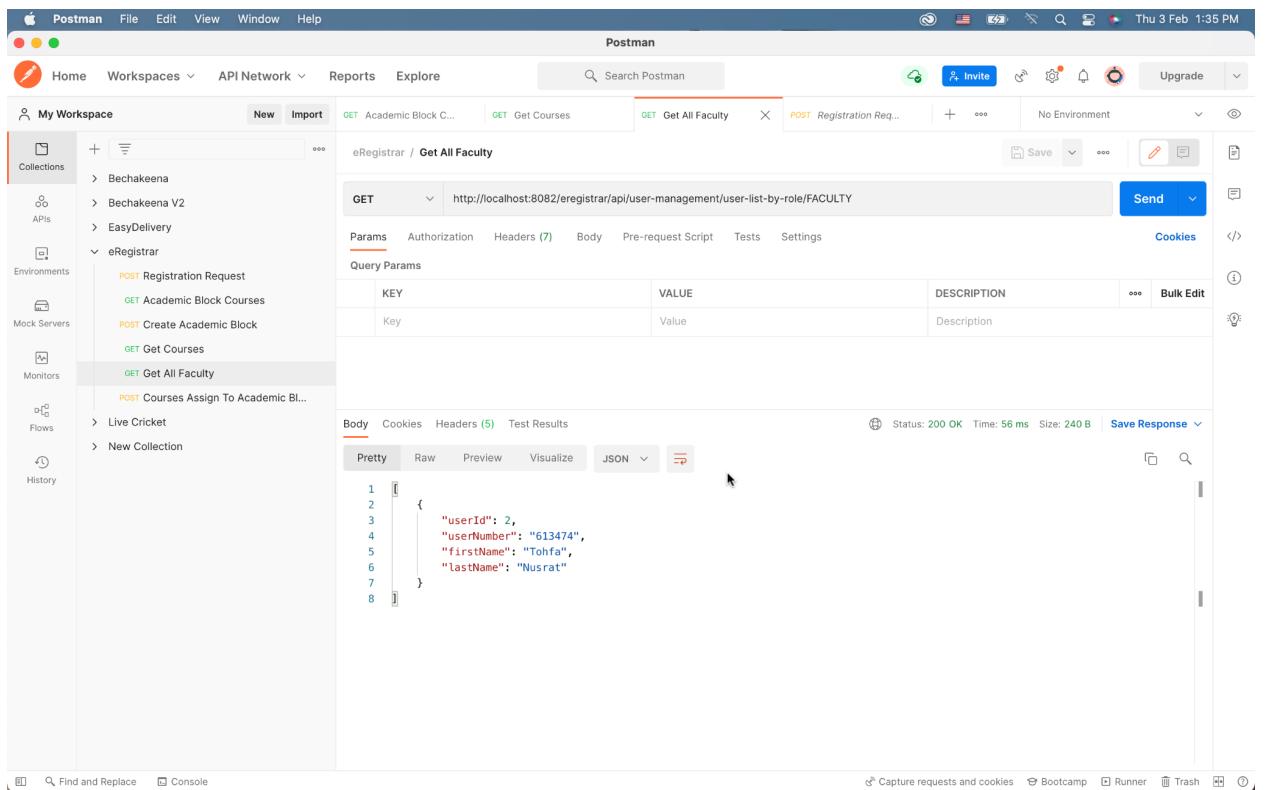


The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists several collections, including 'eRegistrar' which contains 'POST Courses Assign To Academic Bl...'. The main workspace displays a 'POST / Courses Assign To Academic Block' request. The URL is set to `http://localhost:8081/eregistrar/api/academic-block-management/course-assign`. The 'Body' tab shows a JSON payload with course details:

```
1  {
2    "capacity": 45,
3    "academicBlockId": 1,
4    "facultyId": 2,
5    "courseId": 3
6 }
```

The response area shows a small cartoon character and the text 'Click Send to get a response'.

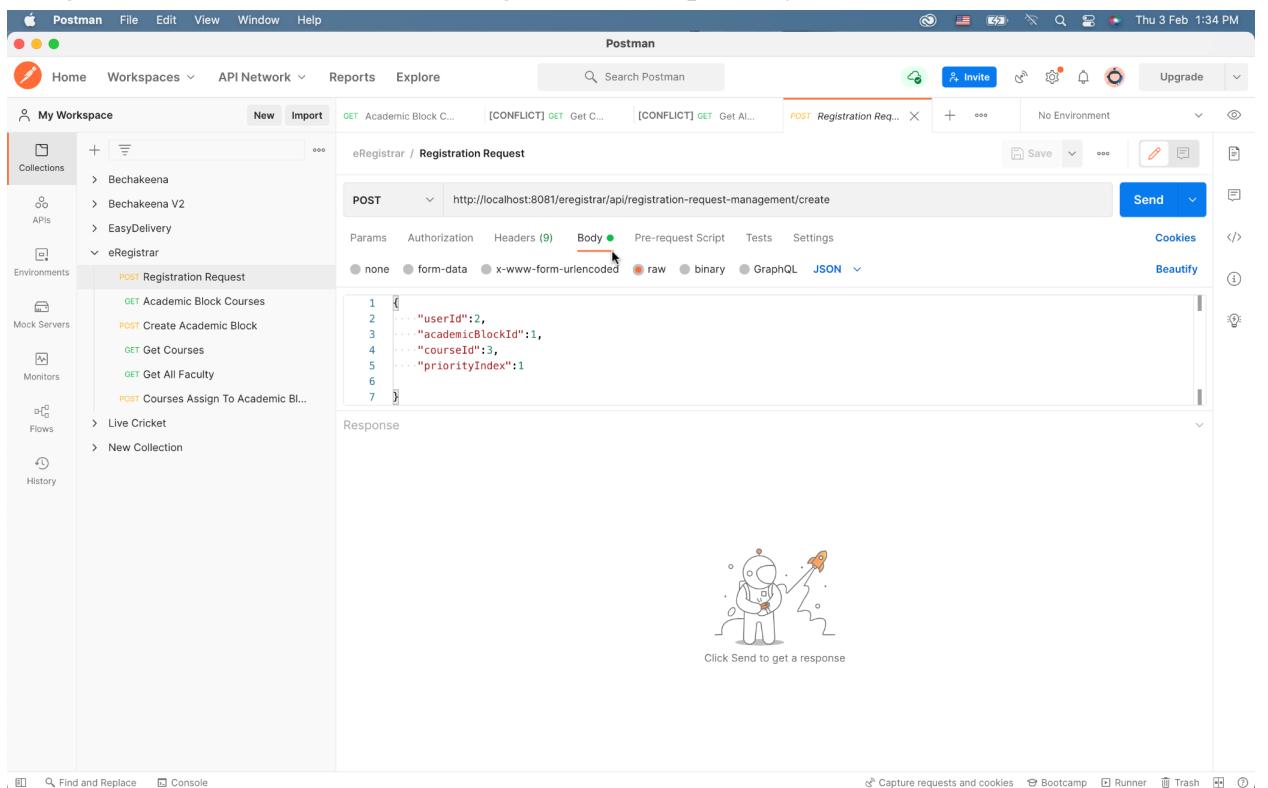
6. eRegistrar Restful API (user list by user role)



The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with collections like 'Bechakeena', 'Bechakeena V2', 'EasyDelivery', and 'eRegistrar'. Under 'eRegistrar', there are several requests: 'POST Registration Request', 'GET Academic Block Courses', 'POST Create Academic Block', 'GET Get Courses', 'GET Get All Faculty' (which is selected), and 'POST Courses Assign To Academic Bl...'. The main panel shows a 'GET' request to `http://localhost:8082/eregistrar/api/user-management/user-list-by-role/FACULTY`. The 'Body' tab is selected, showing a JSON response:

```
1 [ {  
2   "userId": 2,  
3   "userNumber": "613474",  
4   "firstName": "Tohfa",  
5   "lastName": "Nusrat"  
6 } ]
```

7. eRegistrar Restful API (course registration request by Student)



The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with collections like 'Bechakeena', 'Bechakeena V2', 'EasyDelivery', and 'eRegistrar'. Under 'eRegistrar', there are several requests: 'POST Registration Request' (which is selected), 'GET Academic Block Courses', 'POST Create Academic Block', 'GET Get Courses', 'GET Get All Faculty', and 'POST Courses Assign To Academic Bl...'. The main panel shows a 'POST' request to `http://localhost:8081/eregistrar/api/registration-request-management/create`. The 'Body' tab is selected, showing a JSON payload:

```
1 [ {  
2   "userId": 2,  
3   "academicBlockId": 1,  
4   "courseId": 3,  
5   "priorityIndex": 1  
6 } ]
```

8. eRegistrar Thymeleaf Bootstrap UI (user login)

eRegistrar :: Course Registration System

Sign in

Username

Password

Remember password

Login

Tohfa & Anwar :: CS425-SWE

© January 2022

9. eRegistrar Thymeleaf Bootstrap UI (home page)

eRegistrar :: Course Registration System

Course Registration

Welcome, Tohfa Nusrat

Welcome to the Course Registration System



Lorum ipsum is simply dummy text of the printing and typesetting industry. Lorum ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Tohfa & Anwar :: CS425-SWE

© January 2022

10.eRegistrar Thymeleaf Bootstrap UI (create academic block)

The screenshot shows the eRegistrar Course Registration System interface. A modal dialog titled "Create Academic Block" is open in the center. The dialog contains fields for "Block Code" (with placeholder "Enter a string to search for academic block"), "Block Title" (empty), "Start Date" (dd/mm/yyyy format, empty), and "End Date" (dd/mm/yyyy format, empty). Below the fields are "Close" and "Save" buttons. In the background, a table titled "February 2022 FPP Group Registration" lists three academic blocks: Sep-2021, Oct-2021, and Nov-2021. A "Create Academic Block" button is visible at the top right of the main content area.

11.eRegistrar Thymeleaf Bootstrap UI (manage academic block)

The screenshot shows the eRegistrar Course Registration System interface. A table titled "February 2022 FPP Group Registration" displays three academic blocks. The columns are labeled "#", "Block Code", "Block Title", "Start Date", "End Date", and "Action". The "Action" column includes a "Manage Block" link, which is highlighted with a red box. The table rows are: 1. Sep-2021, September 2021, 2021-09-01, 2021-09-30; 2. Oct-2021, October 2021, 2021-10-01, 2021-10-31; 3. Nov-2021, November 2021, 2021-11-01, 2021-11-30. A "Create Academic Block" button is located at the top right. The footer contains the text "Tohfa & Anwar :: CS425-SWE" and "© January 2022".

12.eRegistrar Thymeleaf Bootstrap UI (course assign to academic block & faculty)

The screenshot shows a modal dialog titled "Assign Course To Faculty". Inside the dialog, there are dropdown menus for "Select Course" and "Select Faculty", and a text input field for "Capacity". At the bottom right of the dialog are "Close" and "Assign" buttons. The background of the main application shows a table of academic blocks for February 2022, with columns for "#", "Block Code", and "Block Name". The footer of the application includes the text "Tohfa & Anwar :: CS425-SWE" and "© January 2022".

13.eRegistrar Thymeleaf Bootstrap UI (course list by academic block manage)

The screenshot shows a table titled "February 2022 FPP Group Registration". The table has columns for "#", "Course Code", "Course Title", "Faculty", "Available Seat(s)", "Priority", and "Action". There are two rows of data: one for CS390 (Fundamental Programming Practices) with faculty Md Anwar Hossain, available seats 40, priority Two, and another for CS401 (Modern Programming Practices) with faculty Tohfa Nusrat, available seats 30, and priority "Select Priority". The footer of the application includes the text "Tohfa & Anwar :: CS425-SWE" and "© January 2022".

14.eRegistrar Project Structure

The screenshot shows the IntelliJ IDEA interface with the 'eRegistrar' project open. The left sidebar displays the project structure, including the 'src' directory which contains 'main' and 'java'. The 'java' directory is expanded, showing packages like 'edu.miu.cs.cs425.eregistrar' containing controllers (AcademicBlockController, BlockCourseController, CourseController, RegistrationRequestController, UserController) and models (AcademicBlock, Address, BlockCourse, Course, CoursePrerequisite, Registration, RegistrationRequest, Role, User, ValueObject). Below these are repositories (AcademicBlockRepository, BlockCourseRepository, CoursePrerequisiteRepository, CourseRepository, RegistrationRepository, RegistrationRequestRepository, RoleRepository, UserRepository) and services. The 'resources' directory contains the 'application.properties' file, which is shown in the main editor window. The file contains configuration for the application, including database settings (server.port=8082, spring.datasource.url=jdbc:mysql://localhost/cs-cs425-eregistrar, spring.datasource.username=cs-cs425-eregistrar-sys, spring.datasource.password=test1234), JPA/Hibernate properties (debug=true, spring.jpa.show-sql=true, spring.jpa.hibernate.ddl-auto-create, spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliant, spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect, spring.jackson.serialization.fail-on-empty-beans=false), and other general properties.

```
#=====
#APPLICATION PROPERTIES
#
#=====
server.port=8082
spring.datasource.url=jdbc:mysql://localhost/cs-cs425-eregistrar
spring.datasource.username=cs-cs425-eregistrar-sys
spring.datasource.password=test1234

#=====
#JPA/Hibernate Properties
#
#=====
debug=true
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto-create
spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliant
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jackson.serialization.fail-on-empty-beans=false
```

Thank you!
(Tohfa Nusrat & Md Anwar Hossain)