Sri Lanka Institute of Information Technology

B.Sc. Honors Degree in Information Technology

Final Examination
Year 3 Semester 1 (2022)

SE3040 Application Frameworks

Duration: 3 Hours

May 2022

Instructions to Candidates:

- ♦ Students are required to create a full-stack application
- ♦ Examination consists of 2 programming questions
- ♦ All technical requirements should be satisfied
- ♦ Students are allowed to use internet during the examination
- ♦ This paper contains 5 papers

Specifications

You have been asked to create a timetable management system for a private university. Assume that only lectures are delivered in this University. Lectures will be conducted on weekdays. There are 10 Lecture halls with the capacity of 30 students and lecture hours are 9 AM to 5 PM. Duration of the degree program is 4 years. University admission will happen only once a year.

For a semester you have the following student groups

- Year 01 Students
- Year 02 Students
- Year 03 Students
- Year 04 Students

For the system there are 3 types of users.

- Admin
- Lecturer
- Student

Responsibilities

Admin

1. Create Subjects
2. Allocate timeslots to a subject
3. View student/ lecturer profiles
4. View overall timetable
5. Edit/delete Lecturer/Student Profiles

Lecturer

1. Create account
2. View timetable for the registered subjects
3. View students

Student

1. Create account
2. View timetable for the current semester
3. View lecturer profiles

Question 01 (60 Marks)

Data Models

user
userId
userName
email
password
userType
other

module
moduleId
moduleName
duration
lecturerIds
acedemicYear

Note the following

- Student must come up with the appropriate data model to implement the timetable functionality.
- 'acedemicYear' field in module collection can be used to determine the student groups
- 'other' field in user collection can be used to store user specific data

- 'password' field in user collection must be encrypted.

Backend can be implemented using KoaJS or Spring Boot.

- Create a folder name Backend
- Download the correct zip file from the Netexam into the Backend folder
- Extract the zip file
- If you have selected KoaJS, run 'npm install' before starting the development
- MongoDB must be used to create the database
- 5 marks are allocated for the structure of the directories and sub directories
- 5 marks are allocated for successful package installations in both frontend and backend projects.

1. Backend must be developed with the following requirements. (60 Marks)
    1) There should be an API for each data model:
        i. Users – 18 Marks
        ii. Subjects – 16 Marks
        iii. Timetable – 16 Marks

In each api,

- A separate data access object (DAO) script should handle all database operation for each API. [5 Marks]
- All routes need to be handled separately [5 Marks]

Question 02 (40 Marks)

Frontend must be developed using ReactJS.

- Create a folder name Frontend in the main directory.
- Download the zip file from the Netexam into the Frontend folder
- Extract the zip file
- Run 'npm' install before starting the development
- You can select one user roll from Lecturer and Student to develop.

1. There should be component to register a user. Input fields must be validated in the Register form. (6 Marks)
2. There should be a component to login to the app. Input fields must be validated in the login form. (4 Marks)
3. Once a user login, the app must be loaded into the correct page based on the user type. (5 Marks).
4. After a successful login, based on the user type,
    a. Admin user must be able to (14 Marks)
        i. Create Subjects [4 Marks]
        ii. Allocate timeslots to a subject [2 Marks]
        iii. View student/ lecturer profiles in separate tabs [2 Marks]
        iv. Edit/delete Lecturer/Student Profiles [4 Marks]
        v. View overall timetable [2 Marks]
    b. Lecturer / Student must be able to (6 Marks)
        i. View the relevant timetable [4 Marks]
        ii. View other the other user type (Lecturer or Student) [2 marks]

5 marks are allocated for the structure of the directories and sub directories

---End of the Paper---