TIC2601 Final Exam

Your Name: ________________ **Your ID:** ________________

# of Questions: 19

Date and Time of Exam Creation: Mon, Dec 02, 2024 @ 11:35:04

Total Exam Points: 100.00

Question #: 17

Case Study

The Kent Ridge University (KRU) is planning to develop a new online course registration system (CourseReg) to make it easier for students to register for courses. In KRU, each course carries certain number of course unit from 4 to 16. A course may require students to have completed zero, one or more prerequisite course(s). For example, CS2001 requires CS1001 as a prerequisite. CS2001 is also known as a dependent course of CS1001. Each course can be offered multiple times across different academic years and semesters. A student can register for many course offerings through his or her academic career at KRU.

The data model of CourseReg is depicted in the entity relationship diagram (ERD) provided in the attachment. Study the ERD carefully and answer the following questions.

Questions

The backend of CourseReg will be created using Express.js using Sequelize as the object relational mapping framework and SQLite as the underlying relational database.

a) Suppose you define the required Sequelize models and associations among the models, then apply the forward engineering approach. State the number of database tables that will be generated and explain the role of each table. **(4 marks)**

b) Define the Sequelize model for the Course entity using either the sequelize.define or Model.init approach. Infer the appropriate data type for each attribute. **(4 marks)**

c) Define the Sequelize model for the CourseOffering entity using either the sequelize.define or Model.init approach. Infer the appropriate data type for each attribute. **(4 marks)**

d) Define the Requires and Mounted relationships using the appropriate Sequelize associations assuming that both relationships are bidirectional. **(4 marks)**

e) State whether it is possible to define the finalGrade relationship attribute in Sequelize. If yes, explain how this can be done in Sequelize. If no, explain how you would go about changing the data model so that finalGrade can be defined in Sequelize. **(4 marks)**

(Total 20 marks)

Attachment:

attachment_for_itemid_291155.pdf

Item Weight: 20.0

Question #: 18

Case Study

The Kent Ridge University (KRU) is planning to develop a new online course registration system (CourseReg) to make it easier for students to register for courses. In KRU, each course carries certain number of course unit from 4 to 16. A course may require students to have completed zero, one or more prerequisite course(s). For example, CS2001 requires CS1001 as a prerequisite.

CS2001 is also known as a dependent course of CS1001. Each course can be offered multiple times across different academic years and semesters. A student can register for many course offerings through his or her academic career at KRU.

The data model of CourseReg is depicted in the entity relationship diagram (ERD) provided in the attachment.

The backend of CourseReg will be created using Express.js using Sequelize as the object relational mapping framework and SQLite as the underlying relational database. In the Express.js application, you will require endpoints to perform various data operations on at least the Course, CourseOffering and Student entities.

The app.js file of the Express.js application is also provided in the attachment.

Study the sttachment carefully and answer the following questions.

Questions

a) Define a GET endpoint at the /course path to enable user to retrieve a list of courses for browsing on a frontend web application. The endpoint should accept a courseCode query string parameter, which is optional. If the courseCode parameter is omitted, the endpoint should return all courses in the database. If the courseCode parameter is provided, the endpoint should return only courses whose courseCode contains the parameter value, e.g., if the parameter value is 'CS'', only courses whose courseCode contains 'CS'' should be returned, e.g., 'CS1001'. **(8 marks)**

b) The endpoint that you have just defined in **(b)** is actually not very useful for aiding students in the course registration process. Explain why and describe how you would modify the endpoint in **(b)** or create a new endpoint to better support students in the course registration process. You are **NOT** required to implement the revised or new endpoint. **(4 marks)**

c) Define a PUT endpoint at the /registration path to enable user to register for one or more course offerings in the current academic year and semester using a frontend web application. The endpoint should accept the studentNumber and a list of courseOfferingIds, and register the student accordingly. **(8 marks)**

(Total 20 marks)

Attachment:

attachment_for_itemid_291230.pdf

Item Weight: 20.0

Question #: 19

Case Study

The Kent Ridge University (KRU) is planning to develop a new online course registration system (CourseReg) to make it easier for students to register for courses. In KRU, each course carries certain number of course unit from 4 to 16. A course may require students to have completed zero, one or more prerequisite course(s). For example, CS2001 requires CS1001 as a prerequisite. CS2001 is also known as a dependent course of CS1001. Each course can be offered multiple times across different academic years and semesters. A student can register for many course offerings through his or her academic career at KRU.

The data model of CourseReg is depicted in the entity relationship diagram (ERD) provided in the attachment.

The backend of CourseReg will be created using Express.js using Sequelize as the object relational mapping framework and SQLite as the underlying relational database. In the Express.js application, you will require endpoints to perform various data operations on at least the Course, CourseOffering and Student entities.

The app.js file of the Express.js application is also provided in the attachment.

The frontend of CourseReg will be created using React. Recall that React uses a component-based architecture. A React web application consists of multiple page-level components, which themselves can be composed of other components in a hierarchical tree-like structure.

Study the attachment carefully and answer the following questions.

Questions

a) Create a React component representing an input form that allows user to select the required academic year and semester. Academic year is from "AY 2024/2025" to "AY 2026/2027" and semester is either "Semester 1" or "Semester 2". The component should use appropriate HTML input controls and be able to retain the state of the user's selection. **(5 marks)**

b) Create a React component representing a data table that displays the list of course offerings that are available for a particular academic year and semester for user to select. The data table should be formatted as a HTML table and display courseOfferingId, courseCode, courseTitle, academicYear and semester. The component should use Axios to retrieve the required data from the Express.js backend at an appropriate path with an appropriate HTTP method and with the appropriate parameter(s). **(7 marks)**

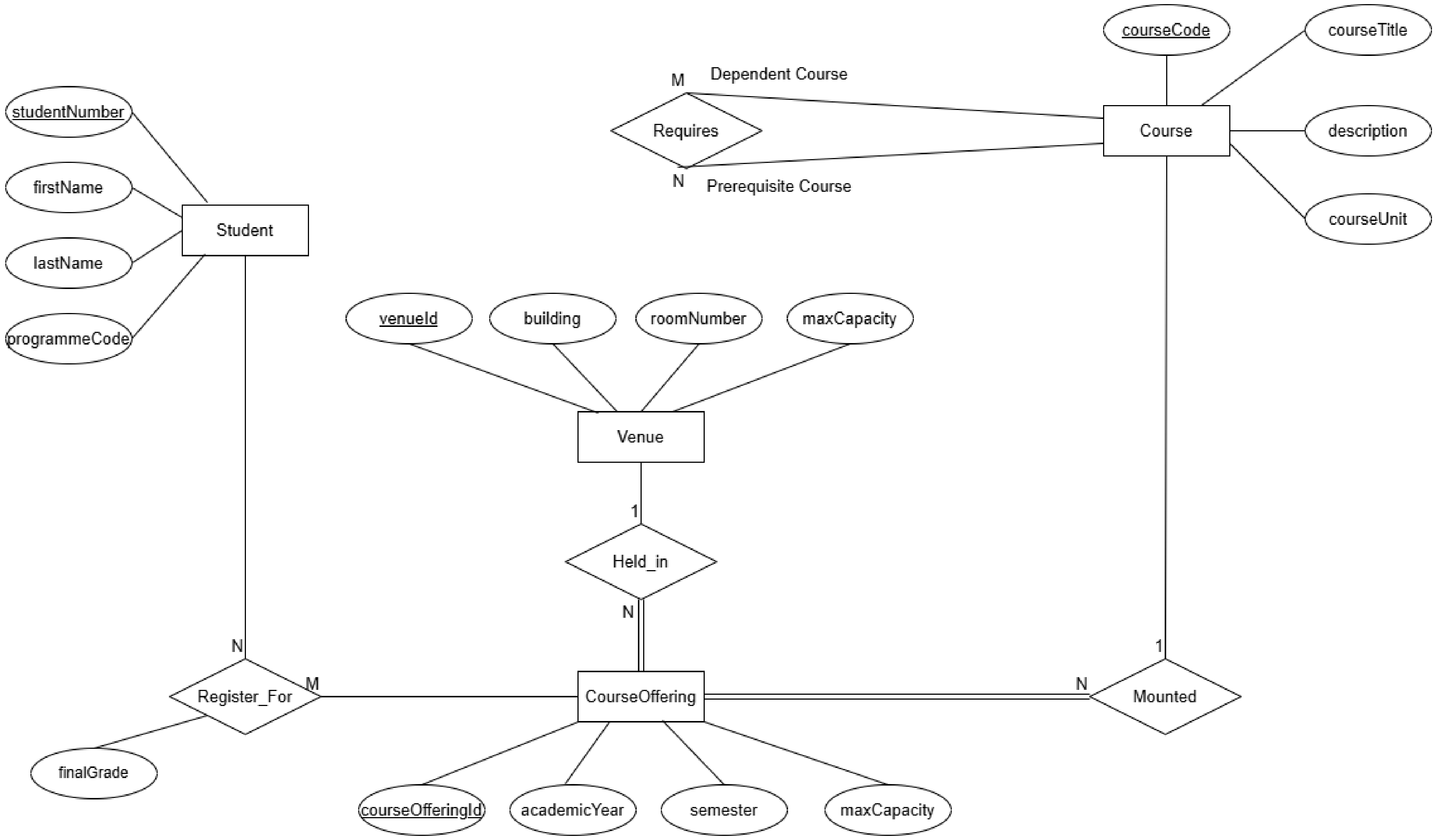
c) Create a React page-level component representing a web user interface for user to register for course offerings. The user needs to select the required academic year and semester, and then select one or more available course offerings. This page-level component must use the components created in **(a)** and **(b)**, and must use an appropriate mechanism to synchronise the states among the components. The page-level component should use Axios to send the required data to the Express.js backend at an appropriate path with an appropriate HTTP method and with the appropriate parameter(s) or request body to complete the course offering registration. **(8 marks)**

(Total 20 marks)

Attachment:

attachment_for_itemid_291278.pdf

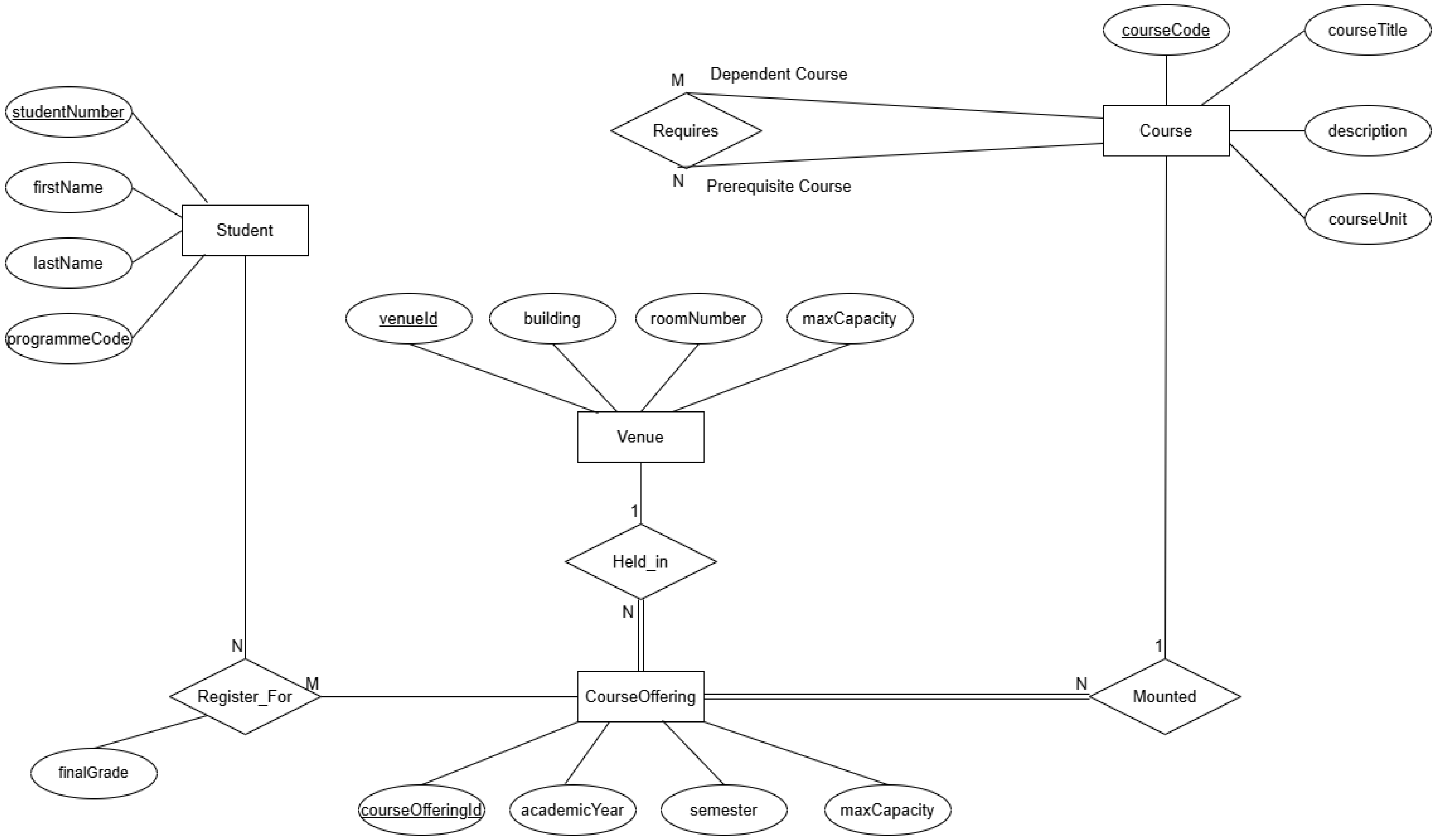
Item Weight: 20.0



ER Diagram for Online Course Registration System

```
const express = require('express');
const cors = require('cors')
const app = express();
const port = 3001;

const models = require('./models');

app.use(cors())
app.use(express.json())

const course = require('./course') ;
app.use('/course', course);

const registration = require('./registration') ;
app.use('/registration', registration);

app.listen(port, function () {
    console.log(`Express app listening on port ${port}!`);
});
```

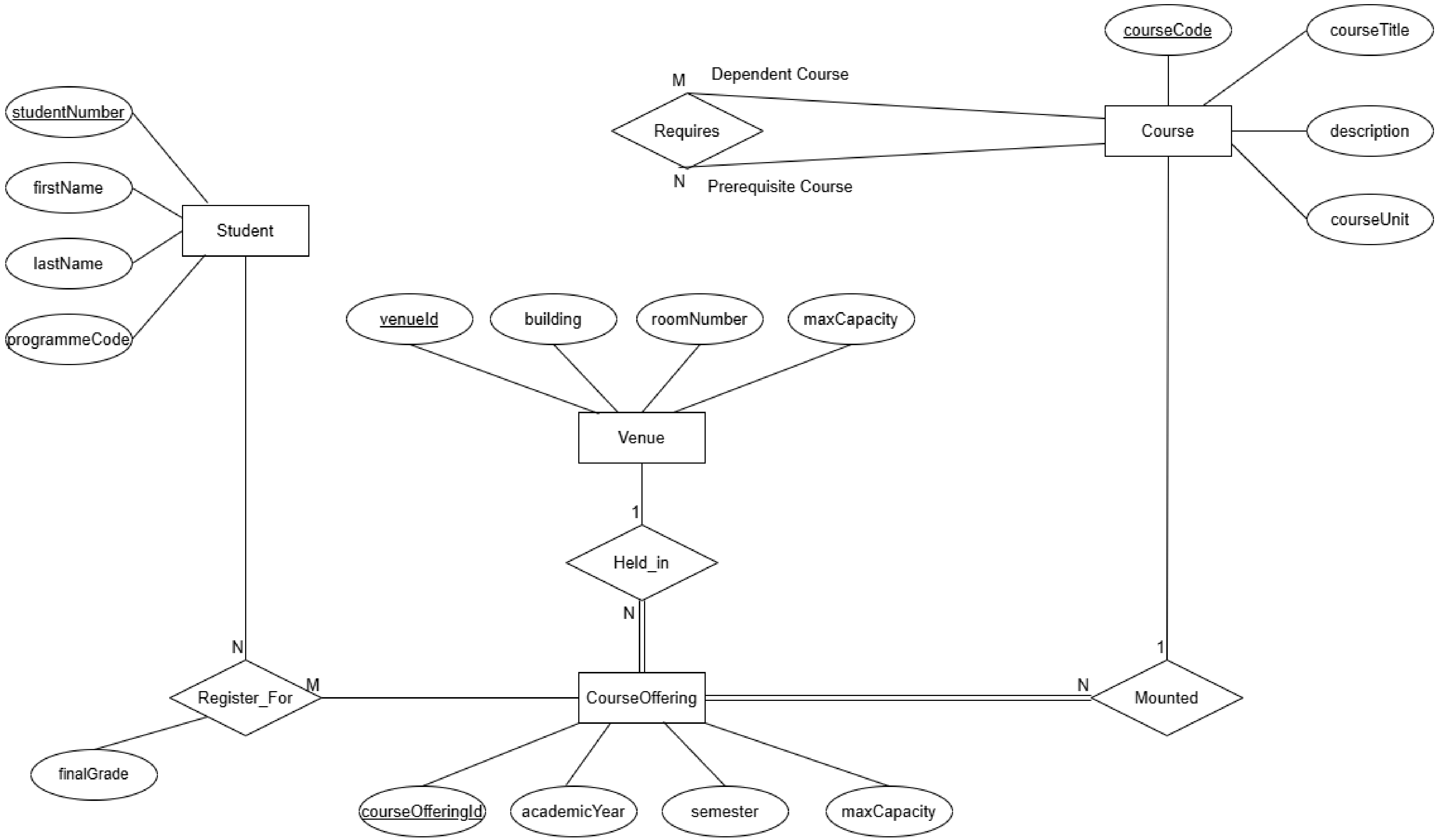
app.js of the Express.js backend for Online Course Registration System



ER Diagram for Online Course Registration System

```javascript
const express = require('express');
const cors = require('cors')
const app = express();
const port = 3001;

const models = require('./models');

app.use(cors())
app.use(express.json())

const course = require('./course') ;
app.use('/course', course);

const registration = require('./registration') ;
app.use('/registration', registration);

app.listen(port, function () {
    console.log(`Express app listening on port ${port}!`);
});
```

app.js of the Express.js backend for Online Course Registration System



ER Diagram for Online Course Registration System