

## Node.js Lab2

- 1- Create eventSystem project folder
- 2- Create package.json file for your project `npm init -y`
- 3- Create MVC folders models, controllers and routers
- 4- install express `npm install express --save`
- 5- create app.js file and call express , then open the server on port number 8080
- 6- Add script to json file `start: "nodemon app"` , then run the server using this script.
- 7- Create 3 middlewares on server
  - a- middleware to write request url and method using `morgan` package (search for morgan on npm )
  - b- Middle ware for CORS.
  - c- general middleware for `not Found` url pathes.
  - d- One Error handling middleware that will catch all system Errors.

You will build small eventSystem that manages events on ITI with speakers and students, the system will have only one administrator with static username and password. Speaker and Student should register first on the system.

Users : administrator(static) , Speaker and Student

### Now start create your routers

- 1- create `authentication router` file, then mange all the following routers
  - a- `/login` - post , to check username and password.
  - b- `/registerStudent`- post , to create student user account
  - c- `/registerSpeaker` – post , to create speaker user account

Speaker Data : `_id(objectID)`, `fullname`, `password`, `email` , `image` (which is string)

Student Data : `_id(number)`, `fullName`, `password` , and `email`

check your router using postman

- 2- Create **SpeakerRouter** that contains
- a- /speakers - get, to get all speakers.
  - b- /speakers/id get, to get speaker by Id
  - c- /speakers post , to add new Speaker.
  - d-/speakers put, to update speaker user Data
  - e- /speakers delete, to delete specified speaker.

check your router using postman

- 3- Create **StudentRouter** that contains
- a- /students - get, to get all students.
  - b- /students/id get to get student by Id
  - c- /students post , to add new student.
  - d-/students put, to update student user Data
  - e- /students delete, to delete specified student.

check your router using postman

- 4- Create **eventRouter** that contains
- a- /events - get, to get all events.
  - b-/event/id – get to get event by Id.
  - c- /events post , to add new event Data
  - d-/events put, to update event user
  - e- /events delete, to delete specified event.

Event Data (title, eventDate, main speaker, speakers which is array of speakers ids, students which is array of students Ids)

check your router using postman