The project repository should be "Accepted/Forked" by team leaders only. One team member (team leader) will add their team members as "collaborators" so they could collaborate and push their daily progress.Create an Express application with the following specifications:

- Allow CORS requests
- Add security headers to the response object
- Given the following environment variable *NODE_ENV*, log all requests to the console when in *development*, and to a file *access.log* when in *production*.

```
process.env['NODE_ENV'] = 'development';
// OR
process.env['NODE_ENV'] = 'production';
```

Provided the following *UserModel*:

```
// download a guest picture and place it in `./images/`
folder: https://pics.freeicons.io/uploads/icons/png/7287311761535956910-512.png

export const GUEST_PICTURE = {
    originalname: "guest.png",
    mimetype: "image/png",
    path: "images/guest.png",
    size: 150
}

const UserSchema = new Schema({
    fullname: { first: String, last: String },
    email: { type: String, unique: true, required: true },
    password: { type: String, required: true },
    active: { type: Boolean, default: true },
    picture: {
        type: {
            originalname: String,
            mimetype: String,
            path: String,
            size: Number
        }, default: GUEST_PICTURE
    }
}, { timestamps: true, versionKey: false });


export type User = InferSchemaType<typeof UserSchema>;

export const UserModel = model<User>('user', UserSchema)
```

Implement the following REST routes:

- POST */users/signup* Signup for a new account (remember to hash the password)
- POST */users/signin* Signin and send back in the response a JWT (contains the user id, fullname, email, and profile picture path)

Users with valid token may:

- POST */users/:user_id/picture* Upload a new profile picture (if they own the account, :user_id === token _id)
- DELETE */users/:user_id/picture* Reset the account picture to the guest picture (if they own the account)
- PATCH */users/:user_id* (with *?action=deactivate_profile*) Deactivate their account (if they own the account).

Provided the following *CourseModel* and the sub-entity of *LectureSchema:*

```
export const LectureSchema = new Schema({
  title: { type: String, required: true },
  description: { type: String, required: true },
  url: { type: String, required: true }
});

export type Lecture = InferSchemaType<typeof LectureSchema>;

const CourseSchema = new Schema({
  title: { type: String, required: true },
  description: { type: String, required: true },
  created_by: {
    user_id: Schema.Types.ObjectId,
    fullname: String,
    email: String
  },
  lectures: [lectureSchema]
}, { timestamps: true, versionKey: false });


export type Course = InferSchemaType<typeof CourseSchema>;

export const CourseModel = model<Course>('course', CourseSchema)
```

Active users with valid tokens can use the application and perform CRUD operations on the *CourseModel*.

Implement the following REST routes for users to perform the following operations:

- POST */courses* Add a new course (user sends course title and description only) (user details to be retrieved from JWT)
- *GET /courses* (*with ?action=all*) List all courses of all users (with pagination)
- GET */courses* (with *?action=own*) List all courses they own (with pagination)
- GET */courses/:course_id* List a specific course by course id
- DELETE */courses/:course_id* Delete a course by course id (if they own the course: the token _id === course created_by.user_id)
- PUT */courses/:course_id* Update a course (title and description) by course id (if they own the course)

And the following sub-entity routes:

- POST */courses/:course_id/lectures* Add a new lecture to a course (if they own the course)
- GET */courses/:course_id/lectures* List all lectures of a course
- PUT */courses/:course_id/lectures/:lecture_id* Update a lecture (title, description, and url) (if they own the course)
- DELETE */courses/:course_id/lectures/:lecture_id* Delete a lecture (if they own the course)

Final Evaluation

The submission *deadline is on Monday at 9:00 PM*. I will meet with you on Tuesday and Wednesday as per the schedule and evaluate your final project submissions.

Good luck, and happy coding!.