

# [Project] - School

**Ouvert le :** mardi 11 septembre 1900, 23:45

**À rendre :** jeudi 11 septembre 2025, 23:46

---

## 1 - Project overview

SchoolInc Ltd is a company which aim to produce the best experience for your student and professor with a web native tool.

Your task is to produce a full application with both GraphQL API and frontend using a framework (React, Vue or Angular) to allow users to access information about relevant data of their student / school life. This include grading, but also courses, class ...

This should also allow professor to display some information about a specific class or student like name, number of student, grading...

The API needs to be GraphQL compliant and use micro services (not one big monolith application)

This project can be done in a group of 2 people.

**This project is about building the API side AND the frontend!**

## 2 - Features

You need to develop multiples functionalities.

*Below the "endpoint" word will be used to describe a endpoint as of the REST mindset - consider it as a capability in GraphQL (or a feature requested)*

- User management solution
  - Create, Read, Update, Delete user
  - User is at least {id, email, pseudo, password, role}
  - You can create a new user even without being logged
  - You can only update yourself (other users cannot update you)
  - You can only delete yourself (other users cannot delete you)
- Authentication need to be setup
  - Different solutions can be used, jwt token is recommended (see tips for more information)
  - All "public" Read endpoints non-logged/anonymous - only grading is protected to see only your grades
  - All Write endpoints need the request to be authenticated (stateless) - except for the register
- A Grade endpoint
  - List all your grades: it should be an array of courses and grades for each courses
  - Capability to only display grades from one courses (or a list of specified courses)
  - Only a professor can create, update and delete a grade
- A Class endpoint
  - List all Classes and allow you to sort mission by name
  - Display one specific class
  - Create, Update, Delete Classes only by professor (cannot delete if a student is affected to a class)
  - Add a student to a class (can only be done by a professor)
- Professor should be able to have some way to display the grading of a student, of a course and of a class including some computed value like the median value or, lowest grade and upper grade.
- Testing is important. You are required to implement a few tests on your project. Focus on testing the core functionalities

### 3 - Tips

Authentication:

- You do not need a full OAuth solution, just a way to generate a token
- You can use PassportJS and the strategy Passport-local-mongoose
- To test a jwt token you can use jwt.io
- You will have a few setup with GraphQL to use it as is

### 4 - Deliverables

You need to submit an archive containing the source code of your project

With a readme for the installation steps and to be able to run your project: **if the project is not runnable a 0 grading can be attributed!**

### 5 - Graded items

Not providing a documentation or basics tests can result on negative points (up to -2)

- User management: 5pts
- Authentication: 3pts
- Grade graph: 3pts
- Professor as admin (read grades from students): 4pts
- Classes
  - From student: 2pts
  - Front professor: 3pts

**Up to 50% of points for each categories can be lost to code quality and/or security. You have to work as you would do it in a company with impact for customers!**

## Résumé de l'évaluation

Groupes visibles

Cohorte Staff


Caché pour les étudiants	Oui
Participants	0
Remis	0
Nécessitant évaluation	0
Temps restant	175 jours 7 heures

Contactez-nous



Suivez-nous



 Contacter l'assistance du site

Connecté sous le nom « Patrick Chevalier » (Déconnexion)