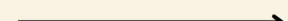


SCHOOL APPLICATION



*creative*  
**PORTFOLIO**

BY TATIANA CLIM



# **Front end**

On the following pages, I will explain how the user can interact with application,  
accordingly to the role of the user.  
And as well, will be presented the flow of the application.



E-Learn

Main Feature

Popular Course

Explore

Blog

Inner Page ▾

LOGIN

REGISTER



# Upgrade Yourself

Anywhere, anytime. Start learning today! And  
Get an Exceptional Education Experience.

SEE DETAIL

GET STARTED



## First Slide

Interdum et malesuada fames ac ante

## Second Slide

Interdum et malesuada fames ac ante

## Third Slide

Interdum et malesuada fames ac ante

The Home Page for my application it will have a header as on the image above, with only two options like Main Feature and Popular Course where when the user will choose those options, will be able to see some statistics like how many teachers are registered on the platform and the most popular courses.



E-Learn

Main Feature

Popular Course

Explore

Blog

Inner Page ▾

LOGIN

REGISTER



# Upgrade Yourself

Anywhere, anytime. Start learning today! And  
Get an Exceptional Education Experience.

SEE DETAIL

GET STARTED



## First Slide

Interdum et malesuada fames ac ante

## Second Slide

Interdum et malesuada fames ac ante

## Third Slide

Interdum et malesuada fames ac ante

On the right upper corner, the user will have the option to signin or to signup. Once he selects one of the options, he will be redirected to the signin or signup page.



Sign up

First Name \*

Please insert a valid name

Last Name \*

Email Address \*

Phone Number \*

Password \*

too short

Confirm Password \*

I agree with terms and conditions.

SIGN UP

[Already have an account? Sign in](#)

Copyright ♦ All rights reserved! Developed by Tatiana Clim  
2022.

The Signup form it will have the same inputs as on the image, and will be validated according with the data inserted.

Once the user will submit the form, the data will be saved on the database, but the user will be able to sign in only after approved by the admin.

Note: This form is only applied for teachers since the students will be created by the admin.

If the user will click on the link below the button, "Already have an account? Sign in" will be redirected to the signIn form.



**Sign in**

Email Address \*

Password \*

Remember me

**SIGN IN**

[Forgot password?](#)    [Don't have an account? Sign Up](#)

Copyright All rights reserved! Developed by Tatiana Clim  
2022.

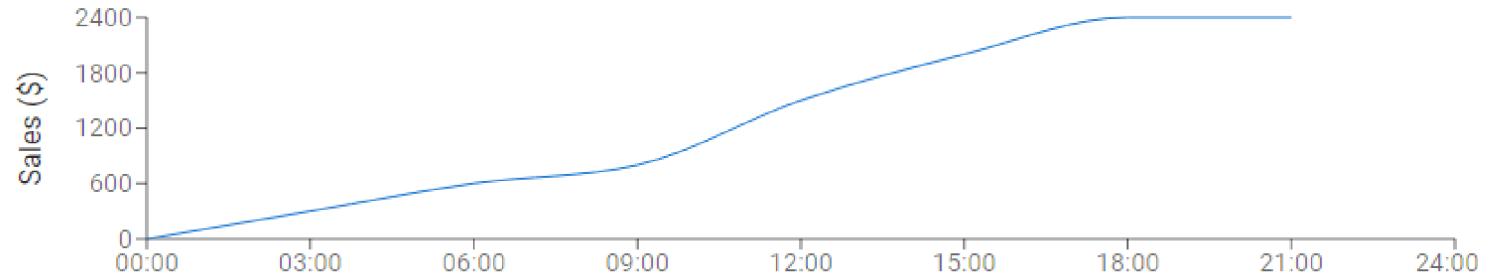
On the Sign In form we have the same link to create an account if the user is not registered, that redirects the user to the sign up form from the previous page.

This form will allow the users(admin, teachers and students) to sign in. If the user exists on the database and the sign in information will correspond, the user will be redirected to the welcome dashboard, according to the role of the user.

## Dashboard

- Dashboard
- Orders
- Customers
- Reports
- Integrations

Today



Recent Deposits

\$3,024.00

on 15 March, 2019

[View balance](#)

Saved reports

Current month

Last quarter

Year-end sale

### Recent Orders

Date	Name	Ship To	Payment Method	Sale Amount
16 Mar, 2019	Elvis Presley	Tupelo, MS	VISA .... 3719	\$312.44
16 Mar, 2019	Paul McCartney	London, UK	VISA .... 2574	\$866.99
16 Mar, 2019	Tom Scholz	Boston, MA	MC .... 1253	\$100.81
16 Mar, 2019	Michael Jackson	Gary, IN	AMEX .... 2000	\$654.39
15 Mar, 2019	Bruce Springsteen	Long Branch, NJ	VISA .... 5919	\$212.79

[See more orders](#)

Copyright © Your Website 2022.

The dashboard page will have the similar design as in the image above. The differences will be the options that each user will have.  
All the pages will have the option to logout. Where the user will be redirected to the home page.

# Dashboard Admin

- Admin will have the option to manage(create/update/delete/view) the students and the classes.
- Admin as well will approve the requests for the new teachers that freshly signed up as well as disable them later .
- Admin will add and remove teacher and students to a class.

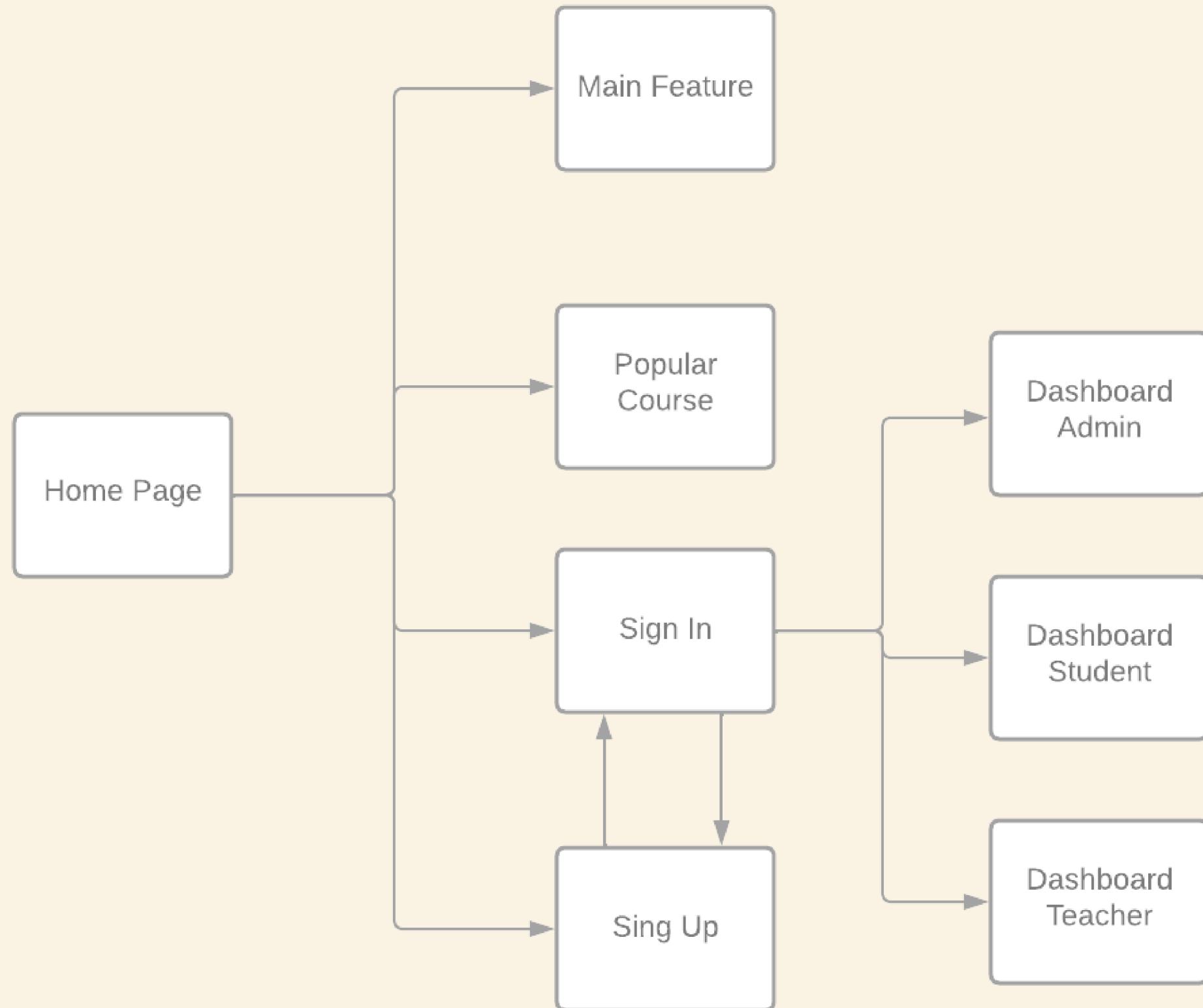
# Dashboard Student

- On the student dashboard, on the left side bar, the student will have displayed all the classes he is assigned to.
- When the student will select a class, he will see all the summaries for that class, and there will be some options available as: mark the presence, submit absence justification, deliver class works and view submitted evaluations.

# Dashboard Teacher

- On the left side bar, will be displayed the classes he is assigned to.
- Once the class is selected, he will have the option to insert summaries, to create a session for users to join, and consult the students attendance.
- Create and submit assignments.
- Create and submit evaluations.

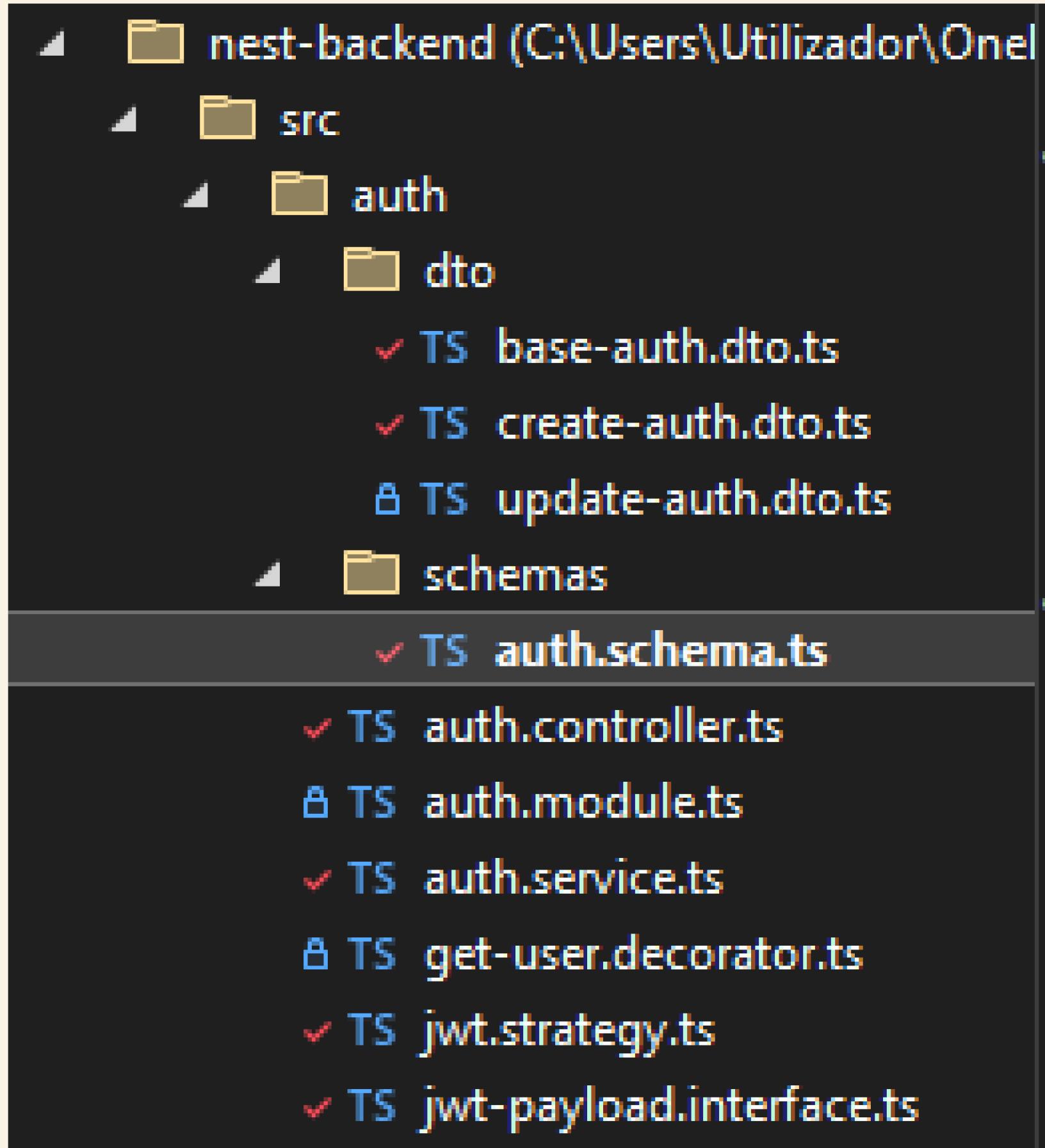
# The flow of the application



No matter on which page the user is, there will be always an option on top bar that will allow the user to return to home page.

# **Back end**

On the following pages, I will explain how the backend application will work.



My backend application will follow the MVC architecture. Also I will determine the DTO Schema.

```
~@Controller('auth')
export class AuthController {
  constructor(private readonly service: AuthService) {}
  @Get()
  async getAllAuth(): Promise<Auth[]> {
    return await this.service.getAllAuth();
  }
  /*
```

The controller will have the CRUD for the users, the CRUD for classes, for summaries and for evaluation.

## **Password**

To encrypt the password in to the database, I will use a bcrypt lib to generate a Salt, and then hash the password with the generate Salt that allows to store the same password with a different output.

## **Token**

When the user is successfully logged in, the data base will return a token, and for this I will use JWT.

# Database

For this application, I will use MongoDB which is a non relational database. This means that it does not use tables, fields and columns structured data concept from relational databases.

# Document databases

Since MongoDB store data in documents, which are usually JSON-like structures, the document off the user will be displayed as on the left hand side.

```
{  
    "_id": "62d9344521b163452cf79e46",  
    "email": "tatiia234sd@gmail.com",  
    "password": "$2b$10$VKtljcRV5Ct.zwUIwyexRuQckLNWvy/vjN4VU11z9SKwdVrZAtKt.",  
    "firstName": "Tatiana",  
    "lastName": "Clim",  
    "phone": "912345678",  
    "status": ACTIVE,  
    "role": STUDENT,  
    "classes": [  
        {  
            "name": "React",  
            "description": "Learn react",  
            "startDate": "21/07/2022",  
            "endDate": "30/10/2022",  
            "summaries": [  
                {  
                    "name": "First Session",  
                    "description": "Create Hello World Project",  
                    "date": "21/07/2022",  
                    "presence": PRESENT,  
                    "assignment": [  
                        {  
                            "name": "First Test",  
                            "description": "Create a country project",  
                            "endDate": "30/07/2022",  
                            "submittedWork": "firstProject.zip"  
                        }  
                    ]  
                },  
                {  
                    "name": "Second Session",  
                    "description": "Create To Do Project",  
                    "date": "25/07/2022",  
                    "presence": ABSENT,  
                    "justification": "justification.pdf"  
                }  
            ],  
            "evaluation": 15  
        }  
    ]  
}
```

# Document databases

```
@Schema()
export class User {
    @Prop({ required: true, unique: true })
    email: string;

    @Prop({ required: true })
    password: string;

    @Prop({ required: true })
    firstName: string;

    @Prop({ required: true })
    lastName: string;

    @Prop({ required: true })
    phone: string;

    @Prop({ required: true })
    role: Enumerator;

    @Prop({ required: true })
    status: Enumerator;

    @Prop()
    classes: Class[];
}
```

The schema for the user, will have fields like email, password, firstName, lastName, phone, role that will be an enumerator[STUDENT,TEACHER,ADMIN] Status will be an enumerator as well[ACTIVE, INACTIVE,PENDING], and an array of classes

```
@Schema()
export class Class {
  @Prop({ required: true })
  name: string;

  @Prop({ required: true })
  description: string;

  @Prop({ required: true })
  startDate: string;

  @Prop({ required: true })
  endDate: string;

  @Prop()
  summaries: Summary[];

  ...
  @Prop()
  evaluations: Evaluation;
}
```

# Document databases

The schema for the Class, will have fields like name, description, startDate,endDate, Summary[] since a class can have many summaries, and one evaluation.

```
@Schema()
export class Summary {
    @Prop({ required: true })
    name: string;

    @Prop({ required: true })
    description: string;

    @Prop({ required: true })
    date: string;

    @Prop({ required: true })
    presence: string;

    @Prop()
    justification: string;

}
~
```

# Document databases

The schema for the Summary, will have fields like name, description, date,presence and justification in case that the student will be absent.