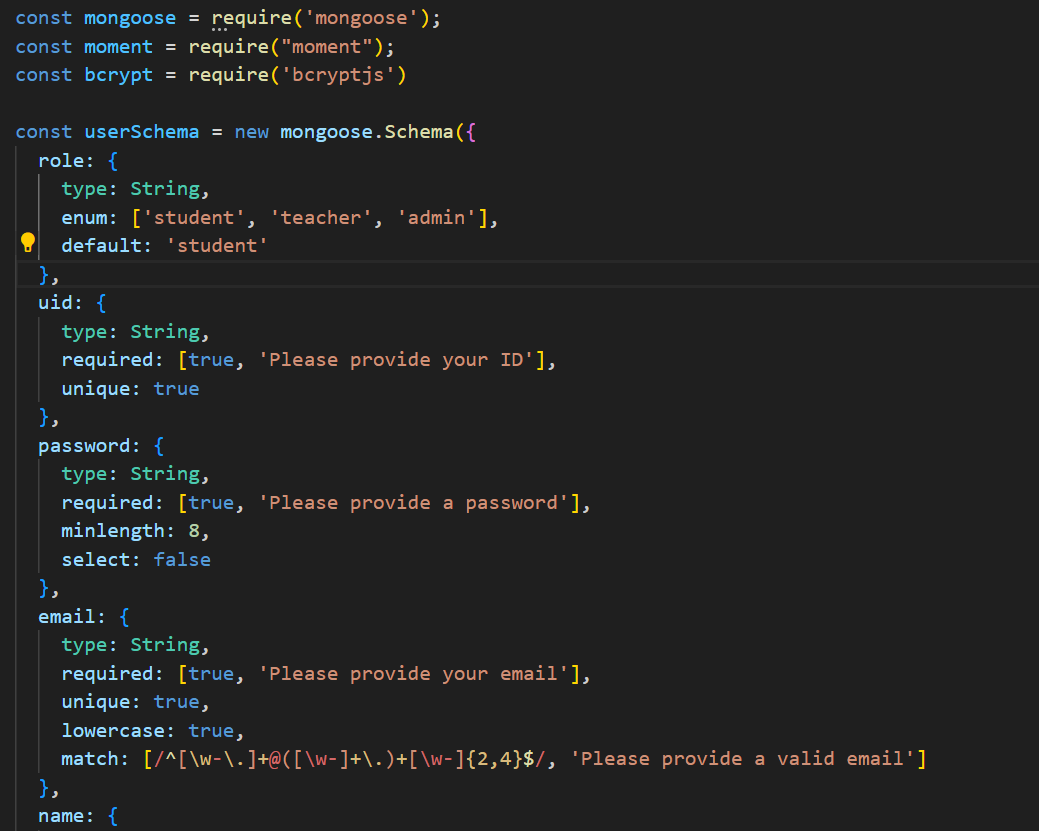
# Testing Plan

## Unit Testing

Our program follows the MVC model. We use the mongoose to create models, use express create the controllers, use ejs to create view. Therefore, unit tests are ideal for testing Models and Controllers.

##### Model Unit Testing

We have three models: Academic Record, Course, User. Three models created by mongoose. One of it as follows:

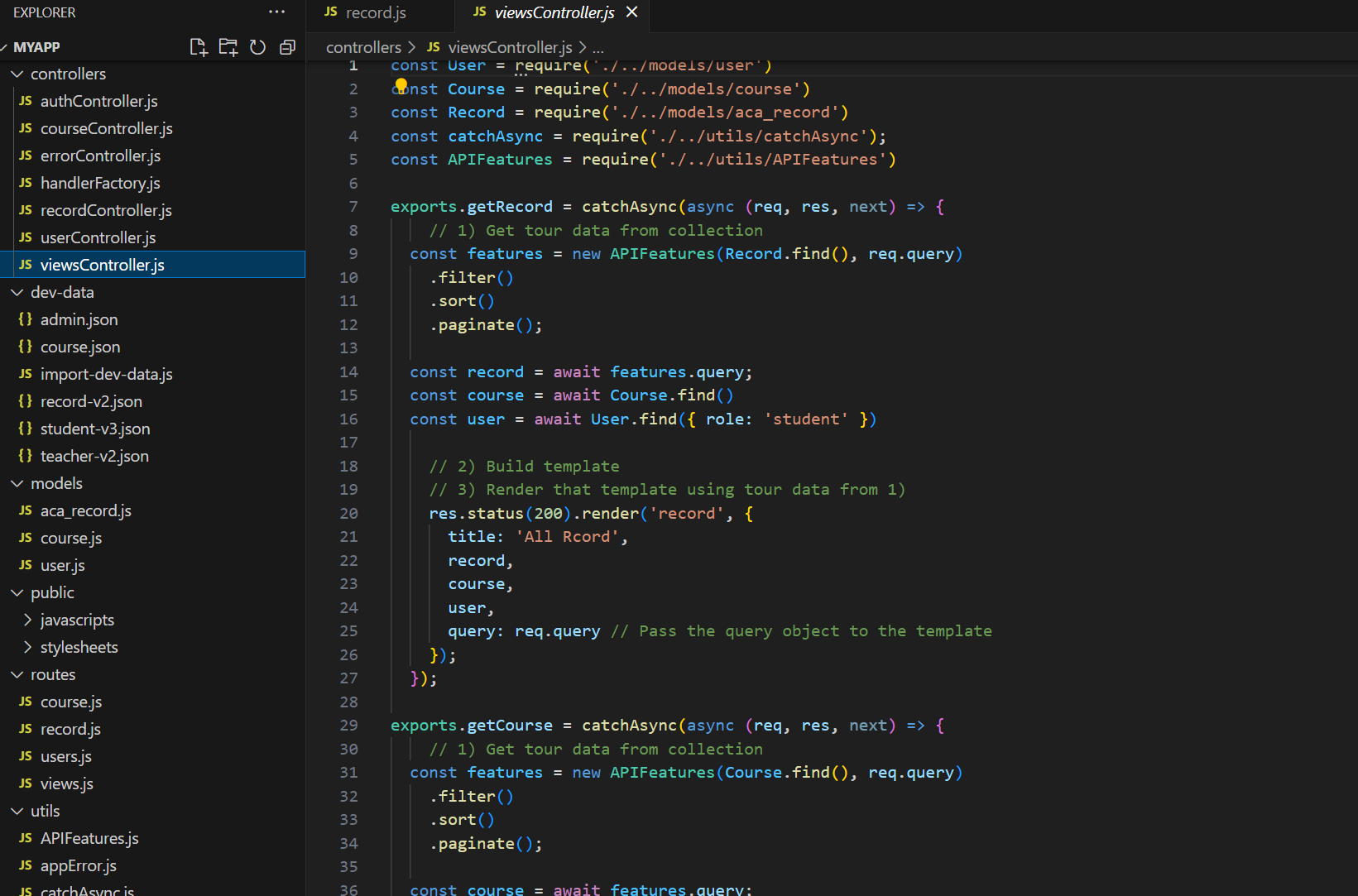


**User Model**

As above figure display, The purpose of testing Models is to test that the definitions are accurate and that the constraints are correct. Specifically, write some unit test cases where you use correct and incorrect data types to see if the data will be inserted into the database correctly.

##### Controller Unit Test

After Model testing, we have to test the most critical part, i.e., the Controllers.Since we are using server-side rendering, the controller's and view rendering are fused together, so the testing process needs to check if the returned templates are as expected. Our project includes seven controllers, one of it as follows:



**Views Controller**

Every controller includes many method, every method will render a html template. The implementation of the test is to call each controller to get the returned Html string, and then use the html parser or regular matching to detect whether the returned string contains the desired data. We plan use api testing library (eg. Super Test).

## System Testing

Since we have tested individual routes in unit tests, but in real cases business is often composed of multiple routes, for example, if a user wants to add data, they must first register and log in. Therefore, we need to test the program systematically, that is, to test the complete process of a particular business, for example, the user wants to delete a course must first register, login, add courses, delete courses.

We will test following functions:

1. User Logout.
2. Update password.
3. Update profile
4. Get courses
5. Delete course
6. Create course
7. Update course
8. Get record
9. Create record
10. Update record

Above functions need to test multiple controllers.

## Acceptance Testing

Finally, we need to simulate the real situation of the program. Specifically, we need to list all the program functions, have team members run the program together, and then test it together. The testing process is to start the web service, open the browser, and perform real user operations. We will test the following features:

1. Register page and register function
2. Login page and login function.
3. Logout function
4. Get courses function, check the layout and data render.
5. Create course page and create course function.
6. Update course page and update course function.
7. Delete course function
8. Profile page render
9. Update profile function.
10. Get records function.
11. Create record function.
12. Delete record function.