For this homework use the rbp version you have that has csrf protection implemented. In my videos I first show how to install mongoDB on CentOS. MongoDB has two parts, a server daemon and a commandline interface, called mongo shell, to interact with the server daemon. You typically run the server daemon as a service, meaning it is running without being attached to a terminal and listens for connections. I show you how to enable the service to start at system boot time. Then, I manually start the service. Then, I use the mongo shell to interact with the server and perform many database operations. I also show you how to use mongoimport to create a database and a collection using a json file.

In the videos about mongoose I show you how you can use mongoose and stay in the JavaScript world to interact with the mongo server. I use mongoose schemas designed for an anatomy lab to illustrate the working of mongoose. I also, show you how it fits in the request handlers of express.

For this homework, you are going to create mongoose schema for the handling homeworks for the School of Computing. The School Schema should have school_name, major_name, minor_name, and an array of courses. Each course in the array of courses should be created using a Course Schema which should have course_name, instructor_name, credit_hours, and an array of assignments. Each assignment in the array of assignments should be created using an Assignment Schema which should have a title, description, assigned_date, due_date, late_due_date, max_points. Add at least one additional scheme of your own.

You should also have a User Schema which should have username, password, academic_standing as its fields.

Create and populate the users collection in the database either using the mongo shell or by using a mongoose based code. Make sure to use csrf protection on all post routes (and optionally on any get route you choose). Write a '/signin' POST request handler and use it to make sure that the username from the request is in the mongo database. If the username is in the mongo database then store authenticated = true in the express-session store. Store any other data you obtained from mongo about the user and what they can access in express-session store. After the client is authenticated, the client should send a POST request to '/getData' and send a date in the body of the request. After verifying that the client is authenticated the server should send the list of homeworks that were assigned or due in two weeks prior and two weeks after the date given in the request body.

Keep all your mongoose models in a models directory. Submit your index.js and models directory and any other js files you create (for example, saga.js in the HomePage component) so that I can run your code. I don't need the entire rbp codebase.