**Lecture 277**

**Resetting Passwords**

* Refer code 01 - resetting passwords - my code
* Views/auth/reset.ejs
* Views/auth/login.ejs 🡪 adding reset link
* Controlllers/auth.js 🡪 getReset method
* Routes/auth.js 🡪 registering new routes

**Lecture 278,279**

**Implementing the Token Logic**

* Send an email to the user with a link , which contains a token which makes sure that the link was sent by the application and is authentic.
* Controllers/auth.js 🡪 postReset method
* Models/user.js 🡪 adding token and token expiration fields
* The token will be saved in the user object in db along with a token expiration time of 1 hour

**Lecture 280**

**Creating the reset password form**

* Refer code 03-creating-the-reset-password-form
* Views/auth/new-password.ejs
* Controllers/auth.js 🡪 getNewPassword method
* Routes/auth.js 🡪 adding routes
* We are extracting the token from the url and verifying with the token from db and also checking if the expiration time is greater than current time.

**Lecture 281**

**Adding Logic to update the password**

* Refer code 04-adding-logic-to-update-password- my code
* Controllers/auth.js 🡪 postUpdatePassword method
* Controllers/auth.js 🡪 getNewPassword method 🡪 passing reset token to the view
* Views/auth/new-password.ejs 🡪 adding hidden field for passwordToken
* Routes/auth.js 🡪 adding new route for /new-password

**Lecture 282**

**Why we need Authorization?**

* Refer notes

**Lecture 283**

**Adding Authorization**

* In the admin products page, we want to display only the products created by that user.
* Refer code – adding authorization
* Controllers/admin.js 🡪 getProducts method 🡪 filtering products based on userid

**Lecture 284,285**

**Adding protection to post actions**

* Refer code – adding authorization
* Filtering products in the admin products page will not alone protect the products of other users from getting deleted or edited. We can still issue the POST requests from some dummy pages. We need to add protection to routes also.
* Controlers/admin.js 🡪 postEditProduct and postDeleteProduct methods