Contents

E_Learn Website

Project idea:

The idea is to create an E-learning platform , where users (students/teachers/graduated people) Can register to the platform and choose to add courses ,and start they learning path (the course) the course contain lessons and each lesson contain an exercise and the exercise need to be done so the user can move to the next lesson , users can track their progress and they will get grades depending on the way they perform.

Requirements:

Accessibility:

The platform need to be accessible from anywhere and for everyone who is registered It should be user friendly (easy to use and understand)

Features: (abstract needs)

Register/Login:

Any user with a Neptune code should be able to register and any registered user should be able to login

Profile/Add courses:

A registered user can access his profile and track his progress and he can access the courses and choose which course to add to his profile

Profile/Remove courses:

A registered user can delete a course in the condition that the admin accept his demand

Practice the course :

The user can practice using an embedded editor and run the code to see the result

Become a teacher :

This features is for teachers , registering as a user and then the user can submit a demand to become a teacher , and this demand will be evaluated by the admin

Teacher /Create courses:

Once the user become a teacher he will find a new feature available which allow him to create courses

Security:

The platform needs to be protected from random access to critical data or resources ,only authenticated users can access their data

Feasibility:

From an abstracted point of view the project is doable by one person, any diving in details can lead to extend the deadline

Projects already in the market (big scale –large scope):

HackerRank

FreeCodeCamp…

Technology used

Node Js:

Node.js cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

Frontend (View ejs):

Embedded JavaScript. It is a simple templating language/engine that lets its user generate HTML with plain javascript. It offers an easier way to interpolate (concatenate) strings effectively.

Css & Bootstrap:

Using also the style sheets and their bootstrap framework

Backend (express):

Express, is a back end web application framework for Node.js, It is designed for building web applications and APIs, Mostly for handling requests

Node js (Passport):

Passport is authentication middleware for Node.js. A comprehensive set of strategies support authentication using a username and password.

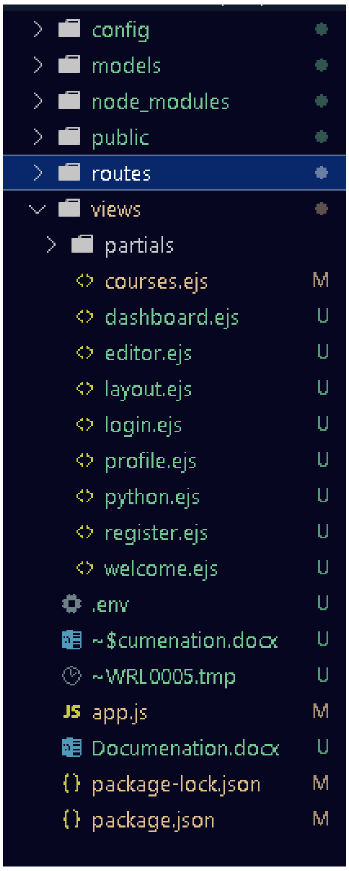
Node js (Mongoose):

Mongoose is a Node. Js-based Object Data Modeling (ODM) library for MongoDB.

Database (Mongo dB):

MongoDB is a source-available cross-platform document-oriented database program.

The project Structure:



```
> config                 ●
> models                 ●
> node_modules           ●
> public                 ●
> routes                 ●
v views                  ●
  > partials
    <> courses.ejs       M
    <> dashboard.ejs     U
    <> editor.ejs        U
    <> layout.ejs        U
    <> login.ejs         U
    <> profile.ejs       U
    <> python.ejs        U
    <> register.ejs      U
    <> welcome.ejs       U
  .env                   U
  ~$cumenation.docx      U
  ~WRL0005.tmp           U
  JS app.js              M
  Documenation.docx      U
  {} package-lock.json   M
  {} package.json        M
```

Config: contains the configurations files of the database and the Authentifiaction

Models: contains all the database collections models that we need

Node Modules: contains all the library used in the project for example (express, passport, mongoose …)

Routes: a JavaScript file used in the sake of abstraction it contains the main routes used in the project for example (user, index, learning …)

Views: ejs files, contains the templates used in the project

.env: For security purposes when uploading to GitHub, this file contains sensitive information and they will not be uploaded

App.js: Js file, the main entry point of the program

Package.json: a json file contains all the information about the installed libraries (name, version ...)