**المدة:** إن شاء الله شهر من الآن (اليوم منتصف كانون الثاني 15/1/2021)

**الـ features:**

* الـ visitor ماذا له؟ الـ logged in user ماذا له؟ الـ enrolled user ماذا له؟ والـ admin ماذا له؟ لكن من هو الـ admin أنه موش مؤلف الكورسات ومعه مؤلف آخر أو اثنين موجودة كورساتهم القليلة على موقعه.
* الكورسات Cards ويجب أن يكون الموقع responsive
* أكثر من Author وهناك فلترة على الـ author وعلى موضوع الـ course وهناك فلترة بالـ search.
* عند الدخول على الكورس قبل شراءه تظهر لك كل المعلومات عن الكورس وفي الأسفل عبارة (join n happy students)
* نموذج شراء الكورس، ورسالة الى المشترك (المنخرط في الكورس) بتفاصيل الـ order والـ discount يتم حسابه عند انشاء الفاتورة بعد معرفة هل المشترك gold أم لا.
* المشترك الـ logged in يظهر له link على الـ navbar هو My Courses عند الضغط عليه تظهر كورساته فقط.
* قابلية تحميل الفيديو
* قابلية فتح الـ pdf file في مربع على الصفحة.
* في البداية تظهر للمشترك بكورس قائمة بأسماء الفصول وفي داخل كل فصل قائمة بالفيديوهات وعند الضغط على أي فيديو مباشرةَ تنتقل الى صفحة أخرى تكون فيها القائمة على شكل sidebar والـ content area لتشغيل الفيديو.

**Features أخرى**

* يُمكن للـ admin إرسال رسائل (هل هي الـ notifications التي سماها موش؟) الى كل المسجّلين بقرب نزول كورس أو وجود فترة تخفيضات أو قرب انتهاء فترة التخفيضات.

--------------

**الـ rights:**

* الـ visitor له التجول ورؤية الكورسات والفلترة ومشاهدة دروس الفصل الأول فقط (فصل الـ Getting Started) فقط وباقي الدروس disabled.
* الـ logged in له ما للـ visitor مع حق الشراء. الـ visitor الذي يضغط على enroll يُطلب منه التسجيل ويظهر له زر ينقله الى الـ register form.
* المشترك الـ logged in يظهر له link في الـ navbar ينقله الى صفحة كورساته وتكون كل الدروس enabled.
* الـ admin الذي يدخل الى الموقع بـ username and password خاصة به يظهر له link في الـ navbar هو admin يكون dropdown فيه manage courses و manage authors، جداول يمكنه من خلالها عمل إضافة أو تعديل أو حذف على course أو على author (مثلاً كورس صار قديماً أو author قد حُذفت كل كورساته لقدمها مثلاً).

نموذج الادخال والتعديل للمؤلف اعتيادي إلا أنه يجب أن يتمكن الـ admin من إضافة صورة للمؤلف من الـ file system فيُخزن المسار وعند الضغط على زر الـ save يجب أن تُخزن الصورة في قاعدة البيانات.

أما نموذج الادخال والتعديل للكورس فهو الـ challenging لأن كل كورس مكّون من فصول وكل فصل مكوّن من دروس.

**خطوات العمل:**

أولاً، إن شاء الله نبدأ بالـ front-end بـ fake services بأقل عدد من الـ properties وبدون الدخول في الـ auth إلا بعد عمل الـ back-end. يتم عمل الـ back-end بعد التأكد من أن كل السباكة تعمل. أول شيء نبدأ بالـ routes.

ثانياً، عمل الأشياء الأكثر تحدياً مثل مسألة أخذ الصورة من الـ file system وخزن مسارها في الـ state لتُؤخذ بعد ذلك وتُخزن في قاعدة البيانات كصورة.

أهم شيء هو نموذج الكورس، وعندي تصوّر عن ذلك فالـ state فيه كل بيانات النموذج (رأس الكورس والـ prerequisites هي مصفوفة وكذلك الـ ableToDos مصفوفة يتم عمل push للمصفوفة بالعنصر المكتوب في input فعند عمل enter يحصل الـ push ويظهر أول li وهكذا. أما محتوى الكورس فهو array of chapter objects والـ chapter object فيه رقم الفصل واسم الفصل والدروس التي هي array of lesson objects).

يكون لدينا input مكتوب فيه "أدخل اسم الفصل"، وعند الضغط على enter تظهر li وبجانبها input مكتوب فيه "أدخل اسم الدرس"، عند كتابة إسم درس وenter تظهر li داخلية nested. أي أن الوصول الى الدرس الأول في الفصل الأول يكون هكذا: course.chapters[1].lessons[1] .

أي أن الـ state لنموذج الكورس سيكون مثل هذا:

state = {

  data: {

    name: "",

    authorId: "",

    prerequisits: [],

    content: [],

  },

  errors: {

    name: "",

    authorId: ""

  }

}

**الـ models:** يمكن أن تكون هكذا

course = {

  \_id: String,

  name: String,

  author: { //hybrid - embedded and standalone

    id: String,

    name: String,

  },

  year: Date,

  category: { //hybrid

    id: String,

    name: String,

  },

  tags: Array,

  info: {

    begin: String,

    prerequisits: Array,

    ableToDos: Array,

    end: String

  },

image: Binary,

  price: Number,

  chapters: Array  //array of chapter objects

}

//the array looks like this:

[

{“ch1”, [{“l1”, “url1”}, {“l2”, url2”}]},

{“ch2”, [{“l1”, “url1”}, {“l2”, url2”}]},

]

}

categories = {

    id: String,

    name: String,

  }

chapter = {

  name: String,

  lessons: Array,       //array of lesson objects

}

lesson = {

  name: String,

  videoUrl: String, //optional

  fileUrl: String, //optional

  //as lesson might be a video and a file OR video only OR file only

}

author = {

  \_id: String,

  name: String,

  address: {

    phone: String,

    email: String,

  },

avatar: Binary,

}

student = {

\_id: String

  email: String,

  password: String,

  enrolled: Boolean,

  courses: Array, //array of courses ids

isGold: Boolean //based on a business logic (gold enrolled with 3 or above

//in config\default.json => configuration setting “minToGold”: 3

}

**الـ implementain:** هذه بعض الأفكار المتفرقة

بالنسبة لظهور الـ Cards بشكل responsive ، لكل صف: 3 على lg و 2 على md و 1 على sm

<div className="container">

  <div className="row">

    {courses.map((c) => (

      <div className="col-lg-4 col-md-6">

        <Card prop1={prop1} prop2={prop2} />

      </div>

    ))}

  </div>

</div>

بالنسبة لظهور الكورس بفصوله ودروسه

<div>

  <ul>

    {course.chapters.map((chapter) => (

      <li>

        {chapter.name}

        <ul>

          {chapter.lessons.map((lesson) => (

            <li>{lesson.name}</li>

          ))}

        </ul>

      </li>

    ))}

  </ul>

</div>