Express.js Exam – Doner Place

Exam rules:

- You have 6 hours – from 9:00 to 15:00
- When you are ready, delete the *node_modules* folder, make sure all dependencies are listed in the *package.json* file and submit your archived project at https://judge.softuni.bg/Contests/824/
- There will be no breaks during the exam but you can go to the toilet or to breathe some fresh air outside **if you are alone**

Problem 1. Home Page & Navigation (20 points)

Get familiar with the provided **html & css** and create an application that stores **products** (Doners), **orders** and **users**. An anonymous user should be able to **register/login** and **logout**. Your **application** should also have **administrators**. An admin can **create** products and change order **statuses**. The **home/main** page should list all **products** in the database by their **categories** and each product should have an **Order** button next to it. Only **authenticated** users can **order** a product. If the user is **not** authenticated **redirect** him to the login page. Admins see additional functionality on the front page (*see provided HTML*).

Problem 2. Create a Product (For admins only) (20 points)

Only **admins** have the ability to **create** a product. Each product has a **unique** category (chicken, lamb or beef), **size** (should be between **17** and **24** cm), **image url** and an array of **toppings** (pickle, tomato, onion, lettuce, hot sauce and extra sauce). The toppings are provided as a comma-separated list that must be parsed on the server.

Problem 3. Place an Order (For authenticated users) (15 points)

All authenticated users can **place** orders for the given **products** by clicking the [**Order**] button at the home page. The button should **redirect** to a page where a user can **customize** his order of a product and can select **all**, **none** or a **few** of the toppings **provided** for each product. If there are **no** toppings, display "**No toppings**".

When [**Checkout**] is clicked, the order is **stored** in the database. Each order has a **creator**, **product**, **date** ordered on, **toppings** selected from the checkbox and a **status** that only **admins** can modify. There are 4 different types of statuses ("**Pending**", "**In Progress**", "**In transit**" and "**Delivered**"). When a user places an order the **default** status should be "**Pending**". The user is then **redirected** to the details of the order that was just created.

Problem 4. My Orders (For authenticated users) (10 points)

Each user can view his **their** orders by clicking [**Order Status**]. The following view should **display** each order of the current **user**. Every **order** should hold information about the **date and time** ordered on, **category** of the product, its **size** and the order **status.** If there are no orders display **"No orders"**. Administrators see additional functionality (*see provided HTML*).

Problem 5. Order Details (For authenticated users) (15 points)

By clicking the [**Details**] button your app should lead to a detailed section of the order which displays the status of your order. The displayed order has a status "**In Transit**" because of the brown **background-color**, while the rest are with a pink filling. Check out the provided **html** and **css classes** to figure out how it is done.

Problem 6. Change Order Statuses (For admins only) (20 points)

Admins can **change** the status of an order. [**All Orders**] link should **render** a view where information about every **order** is listed. At the **rightmost** side of each order there should be a select list to **change** the status. After clicking the [**Change Status**] button, every order that **has** a changed status should be **updated** in the database. Also each order should have a **default** selected status depending on the **current** status of the order.

OTHER REQUIREMENTS

- Use **Node.js** as a web server
- Use **MongoDB** as a database storage
- Use **Express.js** as a routing framework
- You may use whatever frameworks you like