

HƯỚNG DẪN CHUNG KHI LÀM BÀI THI “ONGOING PRACTICAL EXAM”

Lưu ý:

Sinh viên chỉ nộp bài được 01 (một) lần.

Toàn bộ bài làm của sinh viên phải nén lại thành 01 file duy nhất.

Nội dung bài làm và tên tập tin nộp lên hệ thống **TUYỆT ĐỐI** không được bao gồm tên và mã sinh viên (trường hợp sinh viên ghi tên hoặc mã sinh viên của mình trong bài làm, phòng Khảo thí sẽ không công nhận do lỗi đánh dấu bài).

Không sử dụng internet để truy cập các tài nguyên online, các group, diễn đàn, hoặc trao đổi tài liệu, các ứng dụng có tính tương tác như: zalo, teamview, facebook, google drive...

Sinh viên không được sử dụng các ứng dụng được tích hợp AI (như ChatGPT, Codex...).

Sinh viên được sử dụng tài liệu offline có trong máy tính.

Sinh viên chỉ được sử dụng internet để truy cập vào lmsdn nhận đề bài thi và upload bài làm.

Hệ thống sẽ tự động tắt sau khi hết giờ, vì vậy, sinh viên cần phải nộp bài đúng giờ.

SubjectCode	Duration	Ngày thi	Testing system	Yêu cầu đối với bài làm		
				SV được/ không được truy cập internet	Các tài nguyên được sử dụng khi làm bài	Các lưu ý khác đối với bài làm
SDN301m	85	15/07/2023	lmsdn	Được sử dụng Internet để tải đề, cài đặt các thư viện, tài nguyên trên LMSDN và nộp bài;	<ul style="list-style-type: none">- Windows Explorer, zip tools- Visual Studio Code.- MoongDB - Postman- Command Prompt- Trình duyệt (Web Browser) cần tắt các tab/cửa sổ không liên quan, chỉ mở với các URL có liên quan đến localhost: http://localhost:PortXXXX/.... hoặc https://localhost:PortXXXX/....Tất cả các công cụ khác ngoài các công cụ nêu trên (cho dù mở vô tình hay mở không sử dụng) sẽ coi như là vi phạm qui chế thi và sẽ bị 0 điểm PE.	Không nén folder node_modules khi nộp; Bài làm giống nhau sẽ tùy theo mức độ copy để xử lý.

Requirements:

- + Using NodeJS version 16
 - + Only using Internet in order to search information and download packages/modules
 - + Student put the code of the app on 1 zip file, named the file with the meaningful words: **CakeStore.zip** (please avoid using the student's name or student ID and **do not compress node-modules folder**)
 - + Build your backend project that manages Cakes Store. The app allows users to implement the REST API (GET, POST, PUT, DELETE) endpoints /cakes and /cakes/:cakeId to interact with the MongoDB database. Moreover, your app can authenticate users basing on token. It allows to anyone to perform GET operations but only an Admin to perform POST, PUT and DELETE operations.
 - + Student can write the code based on the specific tasks
1. Create the Cakes schema and Topping schema basing on the below template document (Using Mongoose Population and the price schema is be supported with a new SchemaType called Currency) (3 mark)

```
{
  "type": "donut",
  "name": "Cake",
  "price": 2.00,
  "topping":
    [
      {"type": "None", "price_extra": 0},
      {"type": "Glazed", "price_extra": 2},
      {"type": "Sugar", "price_extra": 3},
      {"type": "Powdered Sugar", "price_extra": 2},
      {"type": "Chocolate with Sprinkles", "price_extra": 3},
      {"type": "Chocolate", "price_extra": 2},
      {"type": "Maple", "price_extra": 2}
    ]
},
{
  "type": "donut",
  "name": "Raised",
  "price": 2.35,
  "topping":
    [
      {"type": "None", "price_extra": 0},
      {"type": "Glazed", "price_extra": 2},
```

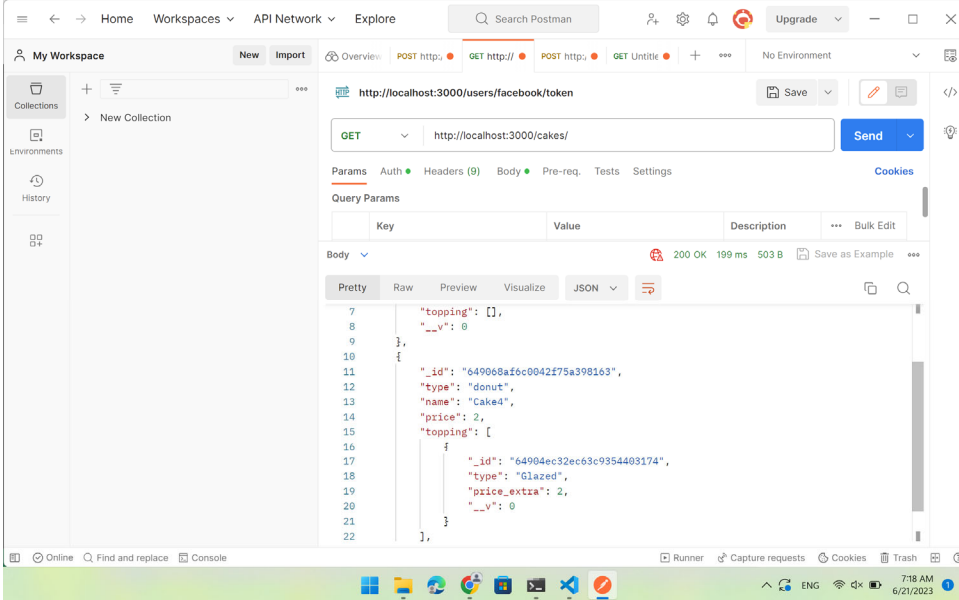
TRƯỜNG ĐẠI HỌC FPT

```
{
  "type": "Sugar", "price_extra": 3},
  {"type": "Chocolate", "price_extra": 2},
  {"type": "Maple", "price_extra": 2}
],
{
  "type": "donut",
  "name": "Old Fashioned",
  "price": 2.7,
  "topping": [
    {"type": "None", "price_extra": 0},
    {"type": "Glazed", "price_extra": 2},
    {"type": "Chocolate", "price_extra": 2},
    {"type": "Maple", "price_extra": 2}
  ]
}
```

2. Do the REST API (GET, POST, PUT, DELETE) endpoints /cakes and /cakes/:cakeId are implemented to interact with the MongoDB database (4 mark)
3. Using token-based authentication in order to allow anyone to perform GET operations and allow only an Admin to perform POST, PUT and DELETE operations (3 mark)

Materials that student must submit to the lsmn system for marking include:

1. ABC_code.zip (70% of marking)
2. The pdf file has **the screenshot** of the application that is tested by Postman. If the app has error, please note "App has error" to the file.(30% of marking)



The screenshot shows the Postman API client interface. The top bar includes navigation links (Home, Workspaces, API Network, Explore) and a search bar. The left sidebar shows 'My Workspace' with 'Collections' and 'Environments' sections. The main area displays a GET request to `http://localhost:3000/cakes/`. The response is a JSON object with the following structure:

```

{
  "topping": [],
  "_v": 0,
  "id": "649068afcc0042f75a398163",
  "type": "donut",
  "name": "Caked",
  "price": 2,
  "topping": [
    {
      "id": "64904ec32ec63c9354403174",
      "type": "Glazed",
      "price_extra": 2,
      "_v": 0
    }
  ]
}
  
```

The status bar at the bottom shows '200 OK', '199 ms', and '503 B'. The system tray at the bottom right indicates the time is 7:18 AM on 6/21/2023.