

Exam Premise:

Create an EJS application that utilizes an random person API to display the address of the person, and to calculate and show the person's current local time.

Exam Details:

The following endpoint can be used to request a random person:

https://randomuser.me/api/

You will receive a response similar to this:

```
{
  "results": [
    {
      "gender": "female",
      "name": {
        "title": "Ms",
        "first": "Gertrude",
        "last": "Gonzalez"
      },
      "location": {
        "street": {
          "number": 4749,
          "name": "Smokey Ln"
        },
        "city": "Manchester",
        "state": "Maryland",
        "country": "United States",
        "postcode": 62719,
        "coordinates": {
          "latitude": 88.6839,
          "longitude": 157.1335
        },
        "timezone": {
          "offset": "+3:00",
          "description": "Baghdad, Riyadh, Moscow, St. Petersburg"
        }
      },
      "email": "gertrude.gonzalez@example.com",
      "login": {
        "username": "angryelephant208",
        "password": "dundee",
        "salt": "24qplY9y",
        "md5": "dd433ac83d71720d165a11ecbca1b4e",
        "sha1": "ae15f3294e162eefcc49d9a238bde96a07600b6",
        "sha256": "7f9c2559bb5f1771c2de8ff85c7410256948484fde9b96c38d2d5ad28c95c9c7"
      },
      "dob": {
        "date": "1991-12-15T11:15:18.683Z",
        "age": 30
      },
      "registered": {
        "date": "2004-08-30T18:15:07.485Z",
        "age": 18
      },
      "phone": "(250) 511-9302",
      "cell": "(938) 508-1886",
      "id": {
        "name": "SSN",
        "value": "531-13-2571"
      },
      "picture": {
        "large": "https://randomuser.me/api/portraits/women/3.jpg",
        "medium": "https://randomuser.me/api/portraits/med/women/3.jpg",
        "thumbnail": "https://randomuser.me/api/portraits/thumb/women/3.jpg"
      },
      "nat": "US"
    }
  ],
  "info": {
    "seed": "dd4704d9a2b2feac",
    "results": 1,
    "page": 1,
    "version": "1.4"
  }
}
```

Use a **JSON formatter** to better visualize your response (for instance [here](#)):

```
{
  "results": [
    {
      "gender": "female",
      "name": {
        "title": "Ms",
        "first": "Gertrude",
        "last": "Gonzalez"
      },
      "location": {
        "street": {
          "number": 4749,
          "name": "Smokey Ln"
        },
        "city": "Manchester",
        "state": "Maryland",
        "country": "United States",
        "postcode": 62719,
        "coordinates": {
          "latitude": 88.6839,
          "longitude": 157.1335
        },
        "timezone": {
          "offset": "+3:00",
          "description": "Baghdad, Riyadh, Moscow, St. Petersburg"
        }
      },
      "email": "gertrude.gonzalez@example.com",
      "login": {

```

UI REQUIREMENTS:

When the page loads, it should make a call to the before mentioned API to retrieve **one random person**. On the UI you will need to display the random person's **first name, last name, and address, consisting of street number, street name, zip code (postcode), city, state, country**. In addition, below the address should be shown the **current local time** for this person. In the API response, you have the data for:

- `timezone":{
 "offset":"+3:00",`

This indicates the offset from UTC. So if the person has a +3 hour offset, if the current UTC time is 2pm, it would be 5pm for that person. Make sure you use UTC to calculate the offset and not the website user's local current time.

Here is a sample mockup for reference:

Personal Data:

First Name:
Last Name:

Street Number:
Street Name:
Zip Code:
City:
State:
Country:

Local Time for the person:

Please follow the mockup structure to display the individual information for the person, as well as their current local time.

Technical requirements:

Use **EJS** as your JavaScript Framework, and utilize the libraries learned in the course such as **express**, and **axios**. You're welcome to use additional libraries as you choose.

You can either create a completely **new EJS project**, or you can use one of the **EJS templates** we have shown in class. If you do use a template, **remove any code** from it that's not being used from the project. **Failing to do so will result in points deduction.**

Extra Credit (10 points):

Style the page according to **Material Design**. You must use **bootstrap** css classes for this, and should not write or modify any css yourself. Your UI should be shown in **two columns**, with First Name, Last Name, and Local Time in the left column, and the rest of the information in the right column.

Submission: Zip your folder **excluding the node_modules** folder and submit it on **Blackboard**. You can submit **as many times as you want**, only the **last submission will be graded**. **Submit often**. The exam automatically **closes at 11.59pm**. **Only code submitted by the deadline will be graded.**

No exceptions for late submissions will be accepted.

Make sure your project **compiles and runs**. If the project doesn't run, you will forfeit points. Do not forget to **comment** and **reference** where needed and appropriate.

Zero tolerance in plagiarism and collaboration. Violation thereof will result in a zero of the assignment for all involved parties, as well as a case report to the department.

Grading Rubric:

Item	Points
UI setup	30
API call	20
JSON parsing	25
Current time calculation	25
Total	100
<i>Deduction 50% if code doesn't compile</i>	(up to -50)