Node JS Server - Assignment #1

This assignment will be done in three parts. Part one creating a basic website that will show the user a “Hello world” message in its homepage. This will lay down the groundwork to build on it. In the second part you will be required to enhance the website to contain multiple pages.

Next, in part three, you will have to improve the server even further. You will read some data from a file and then serve that data as JSON for each URL.

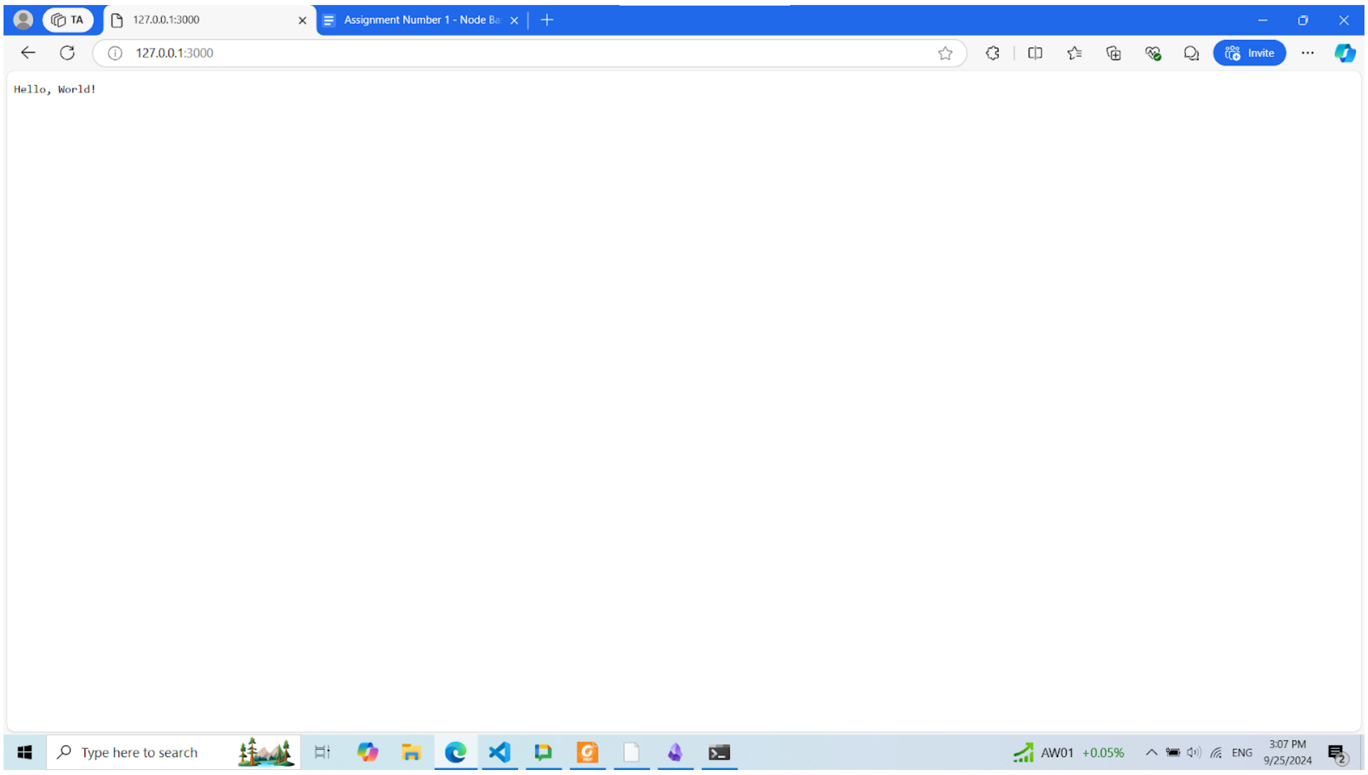
In part 4, you will need to fetch some data from other APIs available on the internet and serve it through your own end points.

You will need to program it in Node JS.

Part I: Basic Server

The user will go through the following steps:

1. Go to 127.0.0.1:3000 in their browser
2. The first page that will be shown will look like the following screenshot

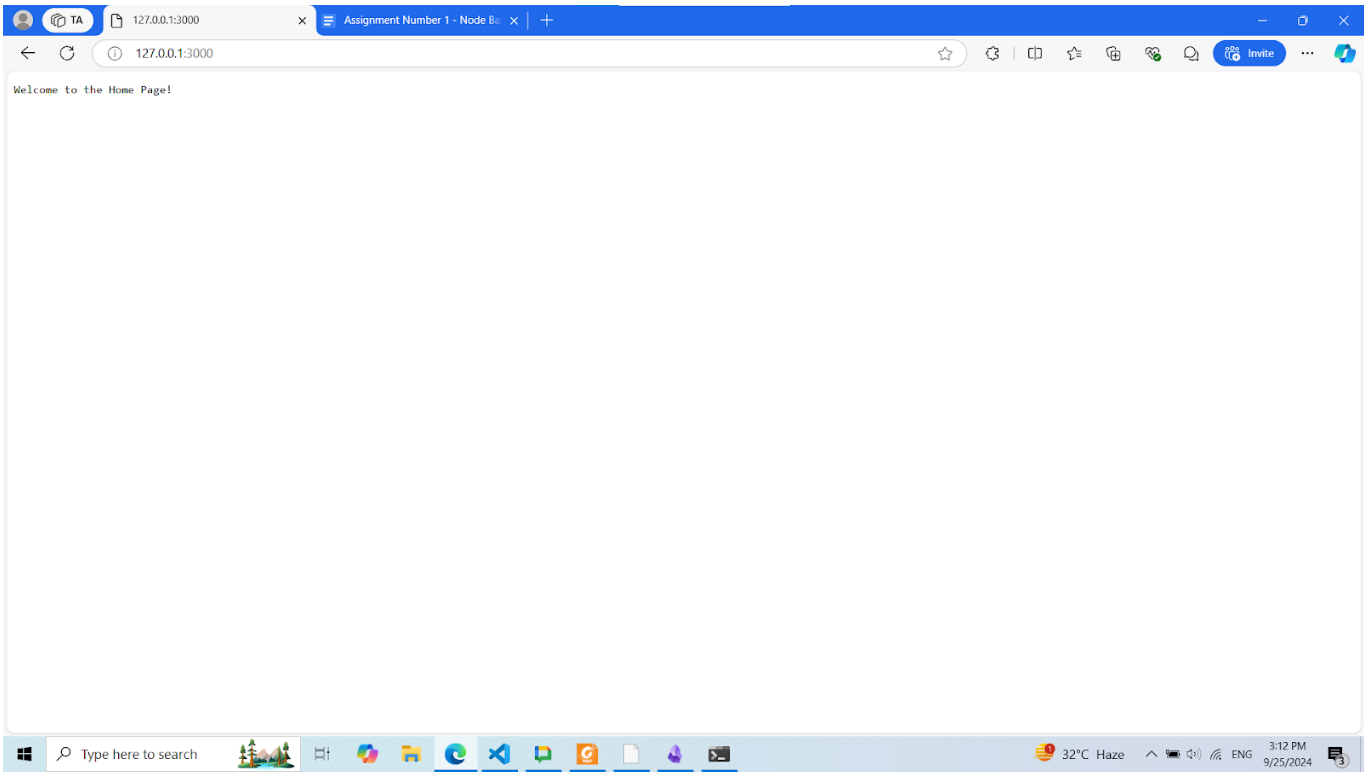


```
Hello, World!
```

Part II: Simple Website

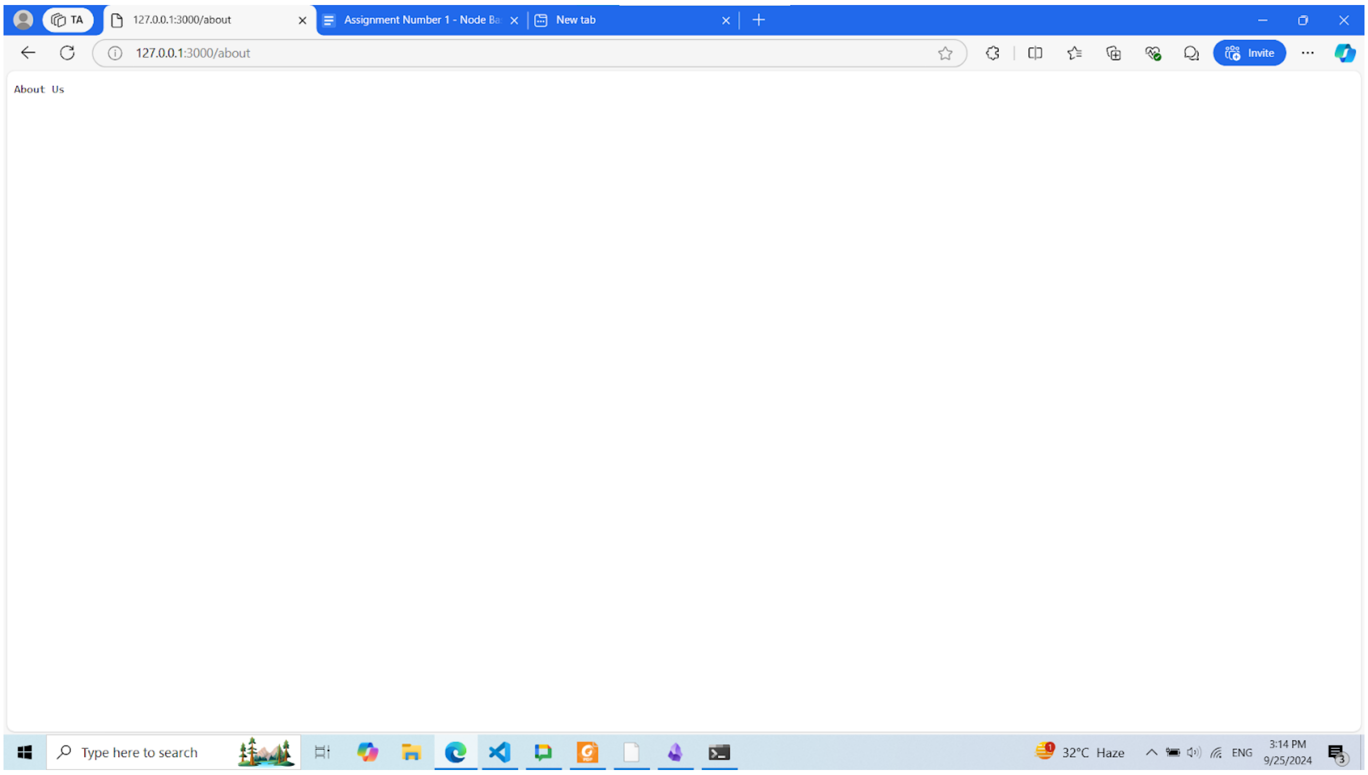
The user should be able to visit the following pages. A screenshot is provided for how each page would look like.

Home Page http://127.0.0.1:3000/



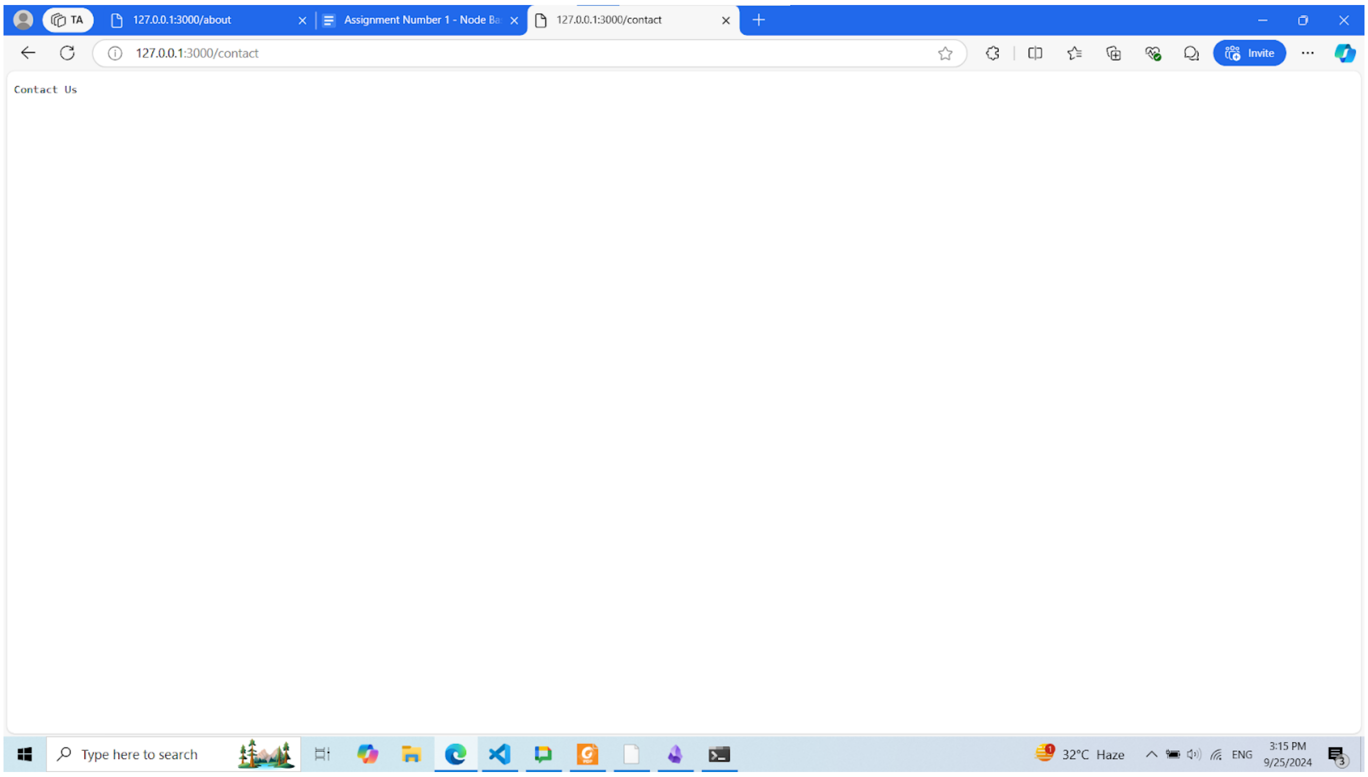
```
Welcome to the Home Page!
```

About Page http://127.0.0.1:3000/about



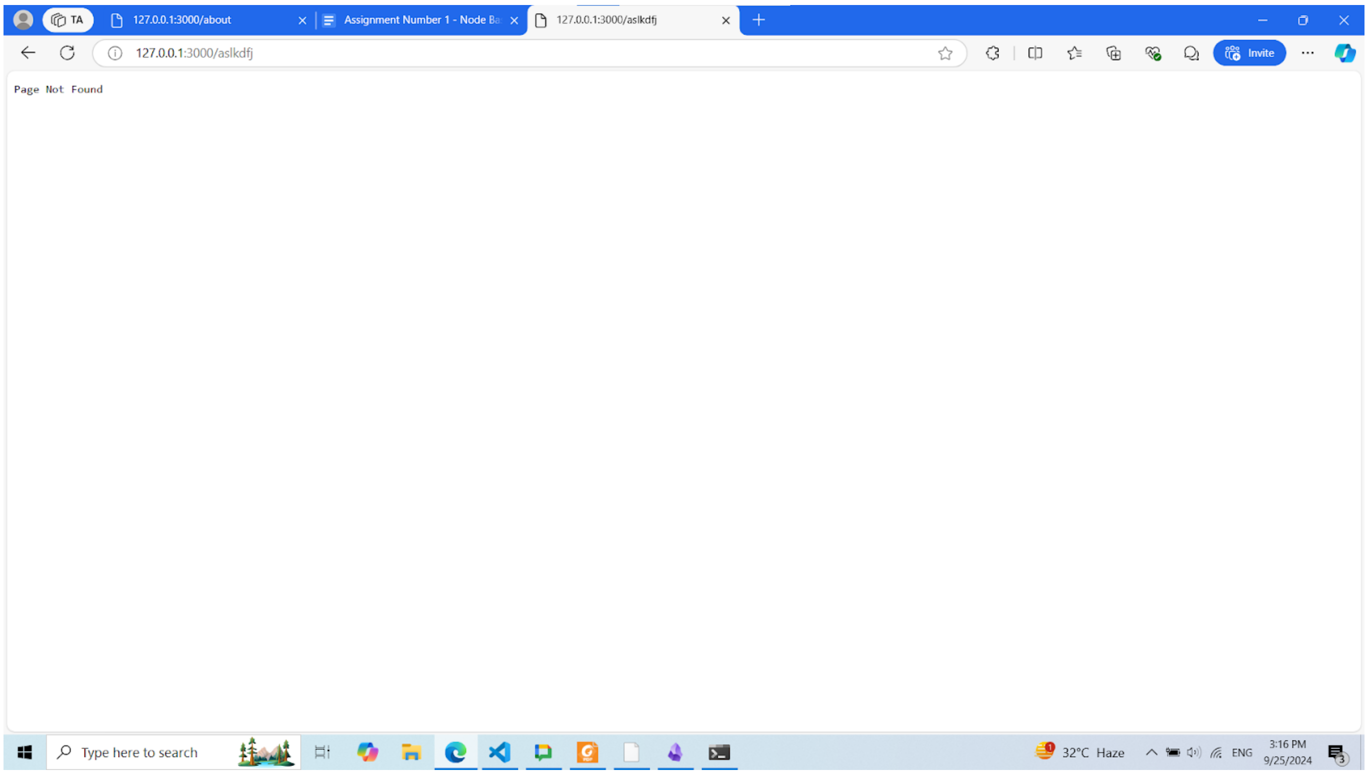
```
About Us
```

Contact Us Page http://127.0.0.1:3000/contact



Contact Us

Not Found Page Visiting any other pages then this will show the following not found message:



Page Not Found

Part III: Serving JSON Data

In this part the following routes should return a JSON response instead of a text.

127.0.0.1:3000`

```
{
    "message": "Welcome to the Home Page!"
}
```

http://127.0.0.1:3000/about

```
{
    "message": "About Us"
}
```

http://127.0.0.1:3000/contact

```
{
    "message": "Contact Us"
}
```

Any other endpoint should return not found message. For example, if you send a request to the following URL: 127.0.0.1:3000/sdlfk

```
{
    "error": "Page Not Found"
}
```

Part IV: Complete RESTful API

In this part you are going change the entire server to work as a REST API. You will need to add the following end points to serve the relevant data in JSON format.

Use fetch API in your Node JS project to fetch data from the following external APIs and then serve the data accordingly through your server.

External APIs to use

1. **Cat Facts API**
    - **Endpoint**: `https://catfact.ninja/fact`
    - **Description**: Provides random facts about cats.
    - **Used In**: `/api/cat-fact` endpoint (with optional `count` query parameter to fetch multiple facts).
    - **API Documentation**: Cat Facts API Documentation
2. **Advice Slip JSON API**
    - **Endpoint**: `https://api.adviceslip.com/advice`

- **Description**: Returns random pieces of advice.
- **Used In**: `/api/advice` endpoint.
- **API Documentation**: Advice Slip API Documentation

3. **Official Joke API**
   - **Endpoint**: `https://official-joke-api.appspot.com/random_joke`
   - **Description**: Provides random jokes including setup and punchline.
   - **Used In**: `/api/joke` endpoint.
   - **API Documentation**: Official Joke API Documentation

4. **Quotable API**
   - **Endpoint**: `https://api.quotable.io/random`
   - **Description**: Returns random quotes along with the author's name.
   - **Used In**: `/api/quote` endpoint.
   - **API Documentation**: Quotable API Documentation

5. **CoinDesk Bitcoin Price Index API**
   - **Endpoint**: `https://api.coindesk.com/v1/bpi/currentprice/BTC.json`
   - **Description**: Provides the current price of Bitcoin in various currencies.
   - **Used In**: `/api/bitcoin-price` endpoint.
   - **API Documentation**: CoinDesk API Documentation

6. **Numbers API**
   - **Endpoint**: `http://numbersapi.com/random/trivia?json`
   - **Description**: Offers interesting facts about numbers.
   - **Used In**: `/api/number-fact` endpoint.
   - **API Documentation**: Numbers API Documentation

7. **Dog CEO's Dog API**
   - **Endpoint**: `https://dog.ceo/api/breeds/image/random`
   - **Description**: Provides random images of dogs.
   - **Used In**: `/api/dog-image` endpoint.
   - **API Documentation**: Dog CEO API Documentation

Hint Use `npm install node-fetch` to install the *node-fetch* package in your project so you can access external APIs.

Your API should have the following end points and corresponding responses.

- **`/api/cat-fact`**:
  - URL: `http://127.0.0.1:3000/api/cat-fact`
  - Optional Query Parameter: `count` (e.g., `?count=3`)
  - Response:

```
[
  { "fact": "Cats have five toes on their front paws.", "length": 42 },
  { "fact": "A group of cats is called a clowder.", "length": 38 },
  { "fact": "Cats sleep 70% of their lives.", "length": 29 }
]
```

- `/api/advice`:
  - URL: `http://127.0.0.1:3000/api/advice`
  - Response:

    ```
    { "slip": { "id": 25, "advice": "Don't eat non-snow-coloured snow." } }
    ```

  - `/api/joke`:
    - URL: `http://127.0.0.1:3000/api/joke`
    - Response:

      ```
      {
        "id": 123,
        "type": "general",
        "setup": "Why did the scarecrow win an award?",
        "punchline": "Because he was outstanding in his field."
      }
      ```

  - `/api/quote`:
    - URL: `http://127.0.0.1:3000/api/quote`
    - Response:

      ```
      {
        "_id": "abcd1234",
        "content": "Life is what happens when you're busy making other plans.",
        "author": "John Lennon"
      }
      ```

  - `/api/bitcoin-price`:
    - URL: `http://127.0.0.1:3000/api/bitcoin-price`
    - Response:

```
{
  "time": { "updated": "Sep 25, 2024 12:00:00 UTC", ... },
  "bpi": {
    "USD": { "code": "USD", "rate": "42,000.0000", ... },
    "GBP": { ... },
    "EUR": { ... }
  }
}
```

- `/api/number-fact`:
    - URL: `http://127.0.0.1:3000/api/number-fact`
    - Response:

```
{
  "text": "42 is the number of teeth dogs are supposed to have.",
  "number": 42,
  "found": true,
  "type": "trivia"
}
```

- `/api/dog-image`:
    - URL: `http://127.0.0.1:3000/api/dog-image`
    - Response:

```
{
  "message": "https://images.dog.ceo/breeds/hound-
afghan/n02088094_1007.jpg",
  "status": "success"
}
```