**Assignment 02**

**HTML, CSS, Bootstrap, Express and Node JS**

**Submission Deadline: See Assessment Dates and Deadlines document on LMS**

**Type: Individual Assignment**

**Submission: Submit your work as one ZIP file on LMS**

**Overview**

* In this assignment, you will be creating a website using HTML, CSS, Bootstrap, express and Node.JS.
* From the website of Pokemon <https://www.pokemon.com/us/pokedex>, you will select any four characters of your choice.

For example, I have selected the Butterfree <https://www.pokemon.com/us/pokedex/butterfree>.

A cartoon butterfly with white wings

Description automatically generated

In the image above, we have highlighted the important information that you will be displaying in your website.

* You will complete this website using the html, CSS, Bootstrap, express and Node.JS to show at least 4 characters of your choice for the Pokedex site.
* For each character, you need to download its image and save its description and other information, highlighted in the image above, in a file.

**Assignment Work and Tasks**

1. Create folder in your computer and name it A2-<Your Student ID> (for example, your folder can be named as A2-12345678).
2. Open terminal or command prompt in this folder and run the following commands:

npm init –yes

npm i express

The first command initializes the Node project in this directory and the second installs the express that you will be using to code HTTP request and its handlers.

1. Open this folder in VS Code.
2. For all the html pages, background color will be set to azure, text size will be set to larger, text color will be set to primary from bootstrap, text alignment will be set to center and bottom margin is set to 30px. Make sure that you are writing these styles in a separate CSS file that you can use in all the pages of your site.
3. Create a HTML file and name it index.html
4. Edit this index.html file and make sure its title is set to ‘A2 – Pokemon’. In this file, you will list at least four buttons of same width and are placed in a borderless and primary table styling from bootstrap. There will be one button for each character with different button style from bootstrap that you have chosen from Pokedex site. You will be using Bootstrap to format these buttons. A sample index.html page is shown below:

A blue rectangle with blue text

Description automatically generated

1. When user of your site clicks on these buttons then they will generate an HTTP request to your server. For example, if use click on Butterfree then it should make a request to your server as follows:

/butterfree

As a response to this request, your server will return a webpage containing the image, description, and information about this Pokemon. We will discuss it later in a new Task.

1. Now create a new file app.js in which you will use implement different HTTP request handler to handle different request from the browser, as given below:

GET /pokemons

GET /<name\_of\_first\_Pokemon>

GET /<name\_of\_second\_Pokemon>

GET /<name\_of\_third\_Pokemon>

GET /<name\_of\_fourth\_Pokemon>

For example, for the index.html page shown above, the HTTP requests are as follows:

GET /pokemons

GET /butterfree

GET /blastoise

GET /charizard

GET /venusaur

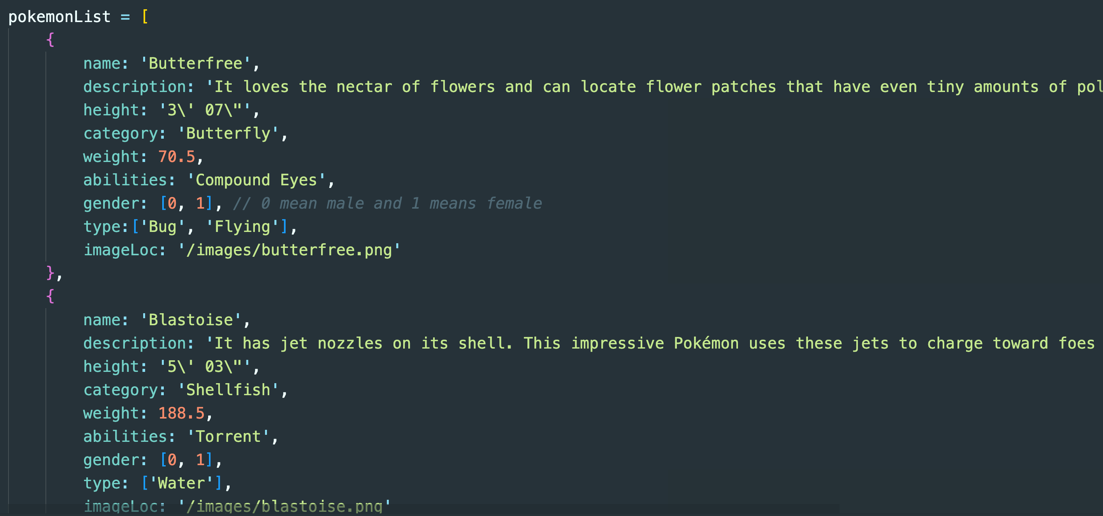
1. You will configure your express app to serve the index.html and CSS file statically to the client and make sure that when you open your website, then it automatically opens index.html page.

You can use the following code:

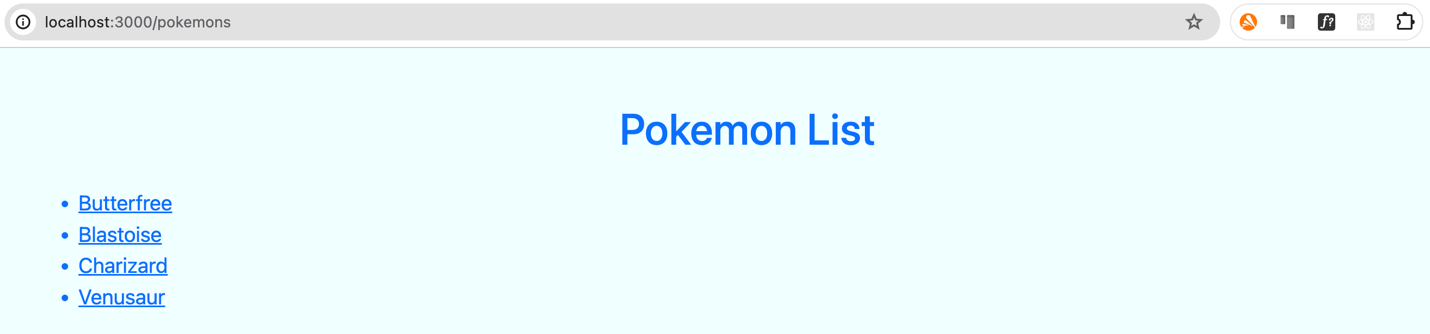
app.use(express.static('./', {index: 'index.html', css: 'a2.css'}));

For more details read: <https://expressjs.com/en/starter/static-files.html>

1. You will create and array (named ‘pokemonList’) of objects in your app.js file that will contain one object for each Pokemon with its information and location of its image. Your array would look like the following:

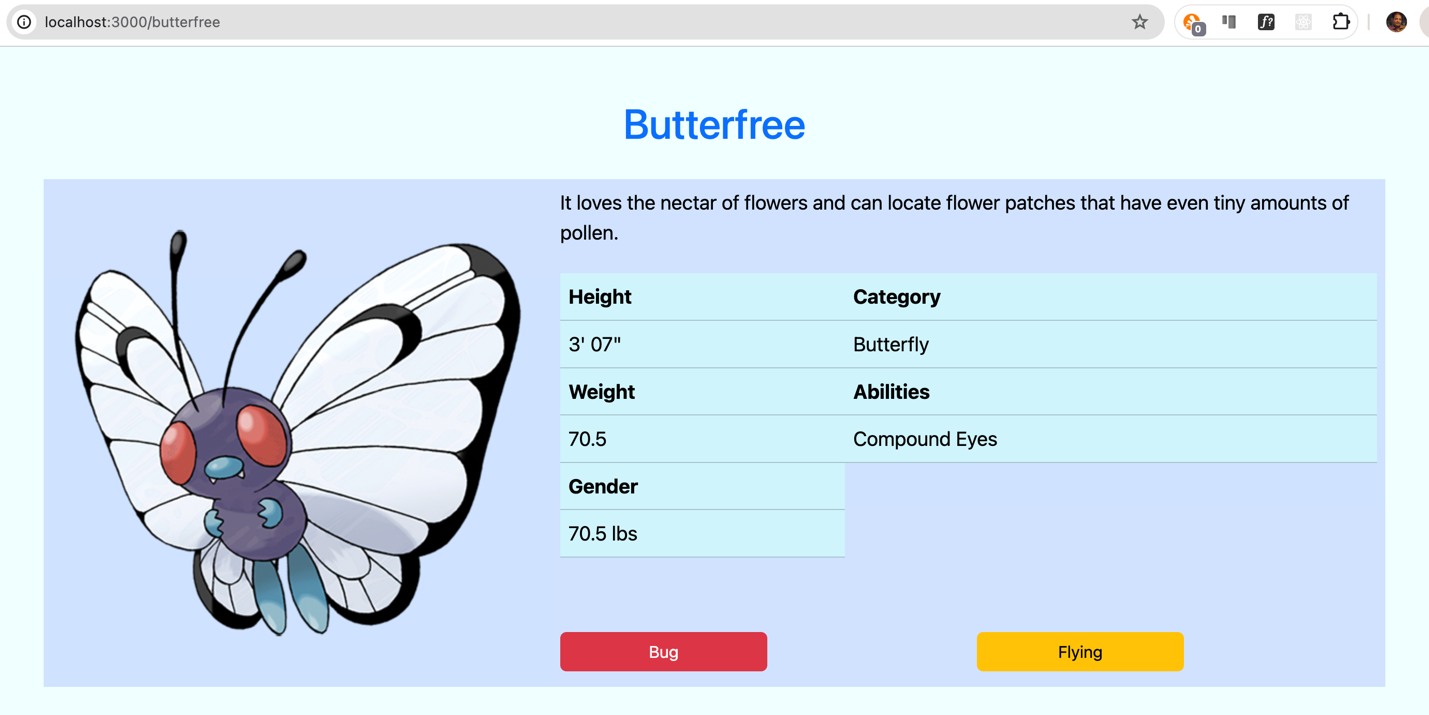


1. Now add GET request handler for URL ‘/pokemons’ in app.js. This will return an HTML string that will show the page as shown below:



This page will show a list of your selected pokemons and the name of each pokemon is linked to ‘/<pokemon\_name>’ GET request like the button in the index.html page.

1. Add a GET request for a Pokemon. For example, add a GET request for URL ‘/butterfree’ and it will return HTML string that will show all the information about the Pokemon (Butterfree) as shown below:



1. Similarly, you will implement the other GET requests for the other three Pokemons.

Beyond these requirements, feel free to include additional CSS to design your pages as you like!