**Name: NISH KAMLI**

**Humber No: N01548012**

**Assignment 2**

**Step 1:**

Create new Node/Express app as follow:

* Create new folder named “Asn2-yourname”.

A black screen with white text

Description automatically generated

* Change directory to “Asn2-yourname”:

A black screen with white text

Description automatically generated

* Open terminal and run npm init

A screenshot of a computer

Description automatically generated

* In the terminal run: npm install express --save

A computer screen shot of a black background

Description automatically generated

* In the terminal run: npm install express-handlebars --save

A computer screen with text on it

Description automatically generated

* Make sure to have the following project structure:

A screenshot of a computer

Description automatically generated

**Step2:** Using the attach file (codenippet.txt), copy/paste the related code to “style.css”, “main.hbs”, “index.hbs”, “error.hbs” and “app.js”.

**Style.css**

A screenshot of a computer

Description automatically generated

**Main.hbs**

A screen shot of a computer program

Description automatically generated

**Index.hbs**

A screenshot of a computer

Description automatically generated

**Error.hbs**

A screenshot of a computer

Description automatically generated

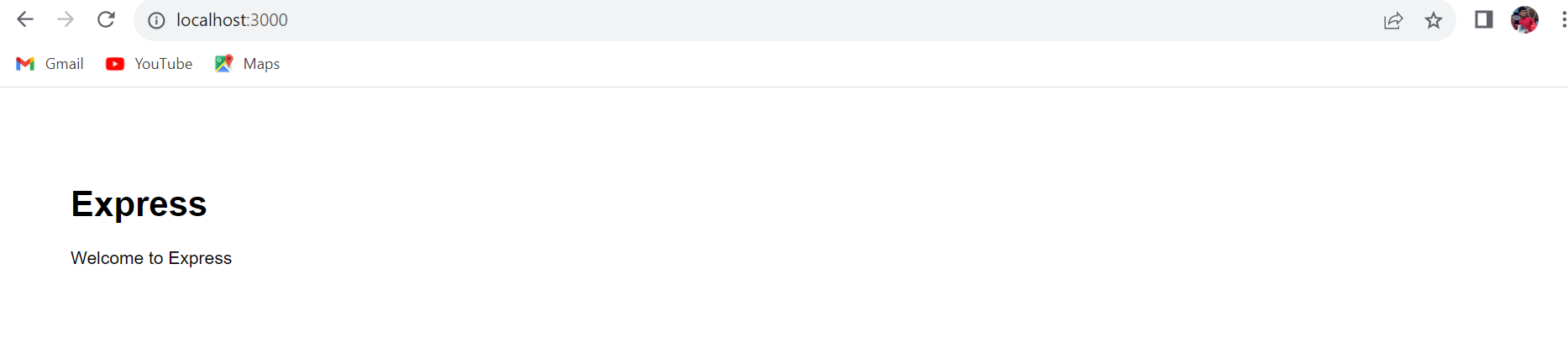
**App.js**

**A screen shot of a computer code

Description automatically generated**

Step 3: Run the application and test it using the following routes:

• http://localhost:3000/



* http://localhost:3000/users

A white rectangular object with a black stripe

Description automatically generated with medium confidence

* http://localhost:3000/notExist

A white background with a black border

Description automatically generated with medium confidence

**Step 4:** Base on your observation, answer the following questions:

1. What is the role of “main.hbs”? How does it being used/called in the app?

the 'main.hbs' file is like a blueprint for all the pages in an app. It sets up the basic structure for how each page should look. Inside this blueprint, there are special placeholders that can change for each specific page:

{{title}}: This is where the title of each page goes. It's like a space that gets filled in with the specific title of the page you're on.

{{{body}}}: This is the main part of the page, where all the content like text, images, or anything you see on a website goes. The triple curly braces mean that anything you put in there will show up exactly as it is, without any changes.

So HTML code entered directly will not be changed. Therefore, by filling in these placeholders with the exact title and content that we prefer for that specific page, we can personalize the primary content and title of any new page you build using this template.

1. What is the role of “index.hbs” and “error.hbs”? How do these files being used/called in the app?

Role of "index.hbs":

It basically Serves as a representation of the application's home page.

Usage in the code: The code res.render('index', { title: 'Express' }); is run when a user navigates to the root URL, represented as '/', in the web browser or the app. This command tells the app to render the index.hbs file and replace {{title}} within that file with the string 'Express'. Consequently, the content of the 'index.hbs' template will be displayed in response to the root URL request.

Role of "error.hbs":

It is designed to show error messages to users who attempt to access nonexistent routes.

Usage in the code: The code contains a wildcard route defined as app.get('\*'...). This route gets triggered when none of the specified routes match the user's requested URL. It then executes res.render('error', { title: 'Error', message:'Wrong Route' });, instructing the app to render the 'error.hbs' template. This replaces {{message}} within 'error.hbs' with the message 'Wrong Route' and displays this error message to the user.

In the code snippet provided:

When a user navigates to the root URL like the one we navigate that is the http://localhost:3000/), the app.get('/') method is called.

When a user navigates to the '/users' route, the app.get('/users') method is called. But in this instance, it only delivers the response'respond with a resource' without rendering a template.

The app.get('\*'...) function is a catch-all route, called when no other routes match the user's request. In this case, it renders the 'error.hbs' template to display an error message for any undefined routes.

The Handlebars templates, 'index.hbs' and 'error.hbs', are rendered using the Express framework's res.render() function, allowing dynamic content to be injected into these templates based on the provided parameters ('title' and 'message').

1. In “app.js”, what is the role of the following line? What happen if you commend it out?

app.use(express.static(path.join(\_\_dirname, 'public')));

This particular code line enables the server to host static files such as CSS or JavaScript from a specific directory called 'public'. By deactivating or commenting out this line, the server would cease to provide access to these static files. Consequently, this action could disrupt the proper presentation of styles and functionalities of scripts on the web pages, potentially causing them to malfunction or display improperly.

1. Try to add proper comments to each line/block in app.js

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

**Step 5**: Modify the layout of the page and add “header”, “content” and “footer” as follow.

* Add your name/std\_id to the header

A screenshot of a computer

Description automatically generated

* Add your email address to the footer

A screenshot of a computer

Description automatically generated

* The other dynamic contents/messages should be displayed in the “content” area.

A screenshot of a computer

Description automatically generated

STEP 6 .1

A screenshot of a computer

Description automatically generated

Step 6.2

A screenshot of a computer

Description automatically generated

6.3

A screenshot of a computer

Description automatically generated

Step 6.4

A screenshot of a car

Description automatically generated

STEP 7 –

A screenshot of a car

Description automatically generated

STEP 8 –

A screenshot of a car

Description automatically generated

STEP 9 –

HIGHLIGHTING UNKNOWN CLASS –

A screenshot of a car

Description automatically generated