Assessment:	**Web Programming 252 CT1**
Assessor	T. Mkwaira
Internal Moderator	-
Total:	30
Duration:	150 Minutes (includes time for uploading etc.)

WEB PROGRAMMING 252 CLASS TEST 1 (&2)

INSTRUCTIONS: GENERAL

- Five minutes have been added for explaining these instructions.
- Keep refreshing AssessmentQ to see the accurate remaining time, or time yourself using a watch.
- No instructions or directives of the invigilator shall be disregarded, if any of the instructions are disobeyed, candidates shall expose themselves to disqualification from the current and future assessments.
- Any updates to the assessment will be posted on HybeFlex and the shared message board
- No explanation of the assessment questions shall be given.
- There will be an even split of marks between this test (CT2), and CT1; in other words, this mark will also be applied to CT1.

INSTRUCTIONS: SUBMITTING YOUR TEST

- Uploading: This assessment must be submitted on AssessmentQ only.
- Naming conventions: When uploading work, ensure naming conventions are followed as instructed. See the saving instructions at the end of this assessment if you are not sure.
- Zipping your work: At the end of this assessment, you should only upload a single zip file containing all your work. See the saving instructions at the end of this assessment if you are not sure.
- File formats: Any work that is uploaded must be in zip format ONLY. Other file types may give problems!
- Optimising your solution & preventing archive errors: Delete *node_modules* before zipping and uploading your final solution.
- Emailing (In case of difficulties during submission): Do not use MS Teams for submissions for whatever reason. In case of technical difficulties, you should submit your assessment to your lecturer with a proper subject line that follows the naming convention WPR252 CT1: StudentSurname StudentName. Your zip file should also follow the same naming convention.
- Reporting technical issues: Any technical issue related to AssessmentQ and HybeFlex must be reported to the invigilator immediately. You may be asked to provide proof of the error, and to follow reporting procedures that may include using the inbuilt reporting tool in HybeFlex.

INSTRUCTIONS: INTEGRITY AND LIABILITY

- Integrity: Observe the Honor Code1. Each candidate is expected to maintain academic integrity. Any form of plagiarism (including self-plagiarism) or cheating that is detected can lead to disciplinary measures being taken.
- Liability: It is your responsibility to ensure that you have saved your assessment correctly! We will not accept excuses and explanations after the assessment.
- Late submissions: Late assessments will not be admitted for marking.

Assessment:	**Web Programming 252 CT1**
Assessor	T. Mkwaira
Internal Moderator	-
Total:	30
Duration:	150 Minutes (includes time for uploading etc.)

INSTRUCTIONS: FORMAT OF THE ASSESSMENT

- This assessment does not include any theory.
- Marks may be reserved for best practices including commenting the important parts of your code etc. and error handling.
- Organise your code well as per the instructions. Further to this, the task must be created using npm init.
- The allocated time includes time for uploading/ saving your solution.
- The mark allocation is subject to change.
- Penalties may be applied for not following the instructions.

Assessment:	**Web Programming 252 CT1**
Assessor	T. Mkwaira
Internal Moderator	-
Total:	30
Duration:	150 Minutes (includes time for uploading etc.)

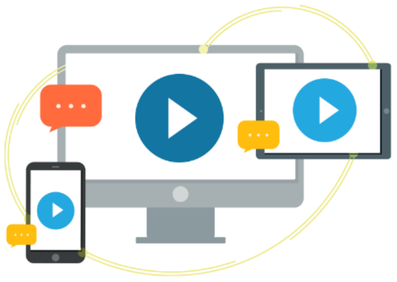
OVERVIEW OF THIS ASSESSMENT.

This practical assessment contains one task which is based on building a server-side application that uses a webpage to create interactions with a user. This assessment considers your skills in HTML (EJS), CSS, Node and Express. Read the instructions carefully; and good luck ☺ !

IMPORTANT: In addition to the questions, I have also included two flowcharts to help you visualize how to build the application.

THE REQUIREMENT FOR THIS ASSESSMENT

When building a web application, you must consider input that should come from the client. This test assesses whether you understand how to build a website that uses a form to submit information to a server, and whether you can perform a basic operation on the server using the submitted information.



**************Your test begins on the next page ***************

Assessment:	**Web Programming 252 CT1**
Assessor	T. Mkwaira
Internal Moderator	-
Total:	30
Duration:	150 Minutes (includes time for uploading etc.)

TASK 1: NODE JS, EXPRESS & EJS [30 MARKS]

Create an application using Node, Express and EJS that allows a person to check whether a word that they enter using the form exists in an array.
Using an array with the following words:

‘foo’, ‘bar’, ‘baz’, and ‘foobar’.

1. Create a server-side application using Node, and Express and ensure that you have installed **all the required dependencies** in addition to express. Remember that:
    1. We will need to use *npm start* to run the application.
    2. You will need to create HTML templates
    3. You will need to parse the request body later in the question.
2. When the server is running, create the following routes: ‘**/**’ and ‘**array-data**’.
3. In addition to the routes above, you should also create a route for processing the form data. Call this route ‘/process-data’. Hint: This route will use the POST method.
4. Create a menu that navigates to the Home page (the ‘/’ route), and also to a page that displays all the values from the array (the ‘/array-data’ route).

At this point you should have a running server that displays the home page and the array data. It is now time to add a form to the application so that we can find if a particular word exists in the array.

5. Create a form that contains only one textbox and one submit button. Use CSS to make the form look good.
6. If the user enters a word in the input box and clicks SEARCH, you should look for the word in the array. If the word exists, print a message back to the user that says, “Word found”, otherwise you should display “Word not found”. You can choose how and where you want to print this message.

Note Use a partial to display the menu.

FLOWCHARTS ILLUSTRATING THE REQUIREMENTS

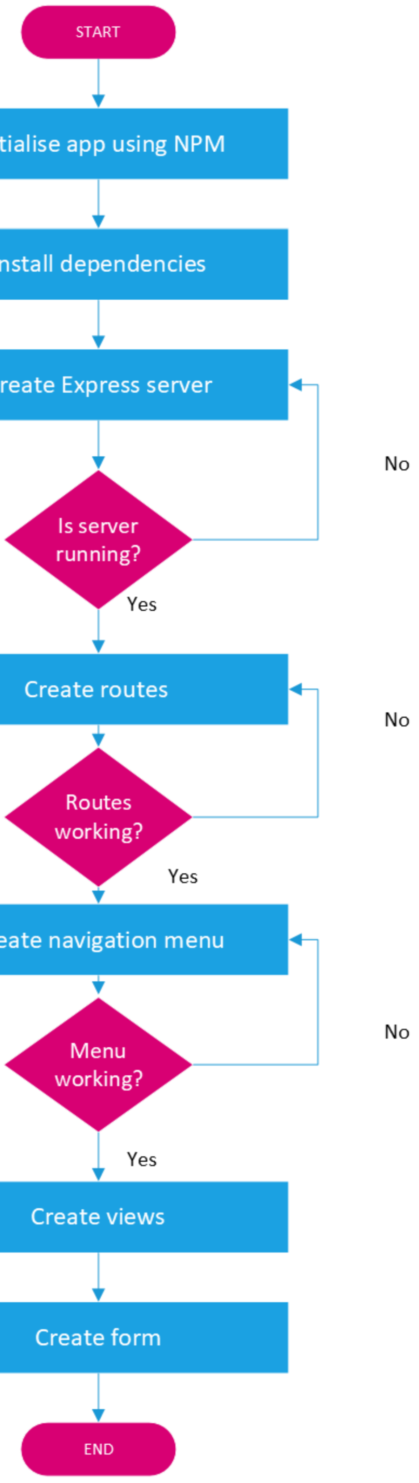


Figure 1: Building the Express server

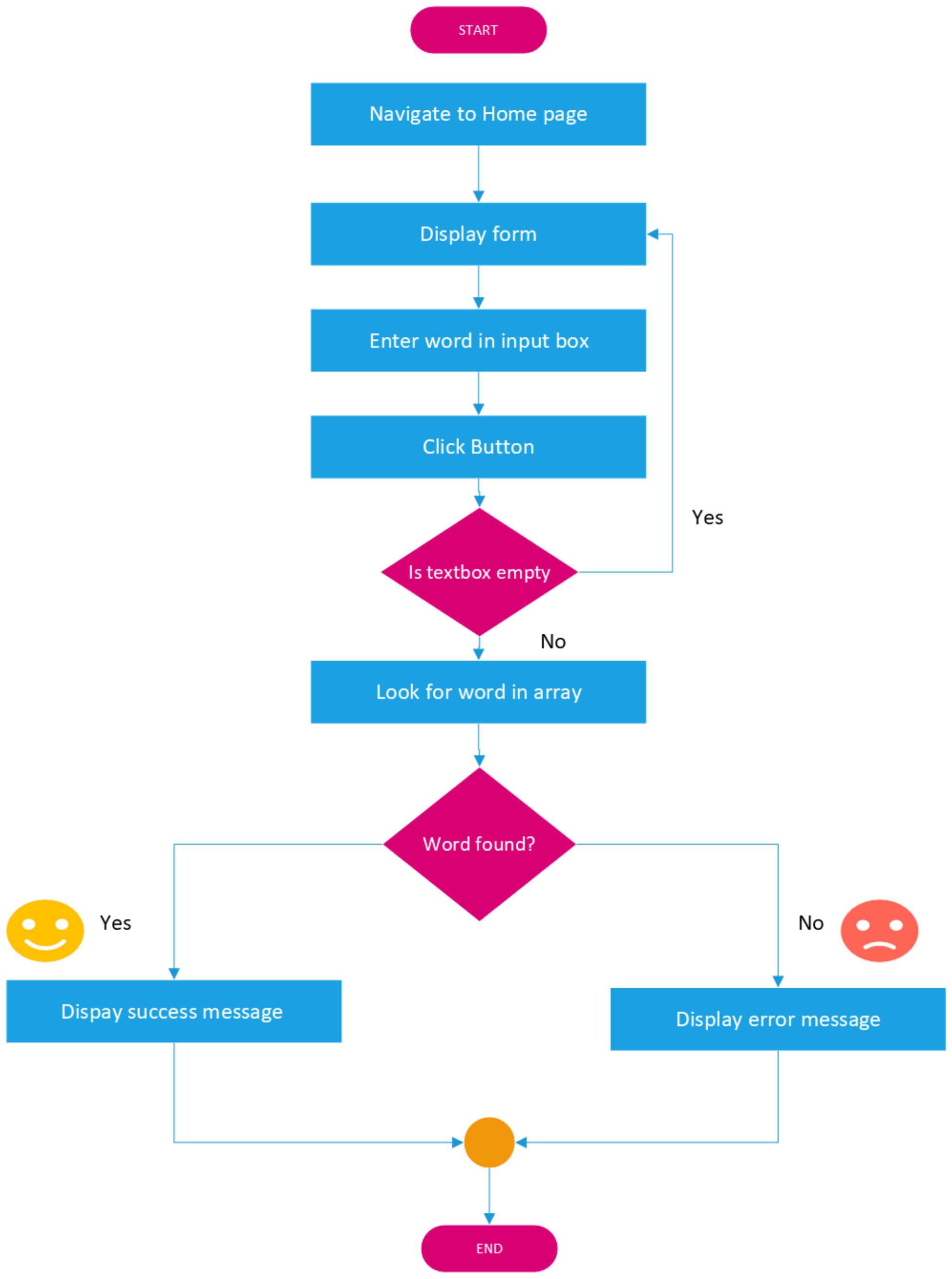


Figure 2:Using the form to look for a word

Assessment:	**Web Programming 252 CT1**
Assessor	T. Mkwaira
Internal Moderator	-
Total:	30
Duration:	150 Minutes (includes time for uploading etc.)

HOW TO SUBMIT: ZIP ONLY! ZIP ONLY!

NAMING CONVENTION

You must submit a **ZIP** file that uses the Campus standard naming convention. Any assessment that a student creates for electronic submission must conform to the following naming convention:

WPR252 CT1: StudentSurname StudentName.zip

The reason for this is to ensure that your assessment is received correctly by your lecturer; this prevents any inconvenience for you as the student and the lecturer.

PACKAGING YOUR SOLUTION

1. Create a folder that follows the naming convention explained above.
2. Move the Tasks folder into the named folder you created in Step 1 (remember to delete node_modules).
3. Finally, create a **zip** archive of the folder you created in Step 1.
4. Upload your zip archive.