|  |  |  |
| --- | --- | --- |
| 5COSC021W Coursework 2 - INDIVIDUAL template 2022\_23  * Use this template to structure the individual part of coursework2. Ensure that the correct information is in each white box. The advice for each box is basic guidance to help you focus your answer. * YOU MUST USE THIS TEMPLATE FOR THE INDIVIDUAL PART OF COURSEWORK 2.  The current size of the boxes is not indicating how much you should write; change their size as you need.When you save the file, put your name and registration number in the file name, eg ‘5COSC003W\_cwk2\_Individual\_Kelly\_Garret\_12345678.doc’.  * A reminder of plagiarism: each group has a different data set and each of you a different task so there is no realistic case where any two students can provide same answers. If you use bits of someone else’s report in yours or if you give your report to someone else to use this will be the academic offence called ‘collusion’. * In order for the tutors to be able to assess your work you must:   - Submit a zipped project folder of the **COMPLETE** working project (i.e. the parts of each group member incorporated in one program, not just your part). If you have not been able to incorporate your part with that of the group, then submit only your part – it should be able to run though by itself.  -          Make sure that the submitted project will run on the Universities computers.  -          Make sure that the project folder should contain all files necessary to run the program e.g. excel files etc.  -          Make sure that file I/O code does not use absolute file paths.  -          Make sure that the submission contains all usernames and passwords necessary to test the program. | | |
| **Surname** | Stephens | |
| **Forename** | Mason | |
| **Registration No:** | W1781585 | |
| **By submitting this coursework you agree to the following:** | | |
| I confirm that I understand what plagiarism is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged. | | I confirm |
| **List here the team name and the other members of your group** | **Gheorghe Pop**  **Kliti Noka**  **Philip Oduro**  **Sanjeeban Niranjankumar** | |

|  |
| --- |
| Code – Code functionality – individual element and integration to application (20 marks) |
| **Guidance:**   * Include here the **names of the files** that make up the individual element of your application. * Explain what each file does and the functionality of the code. Detail whether all the code in the file is your own product or a product with other team members. Include the names of the other authors where applicable. Authorship information should also be commented at the beginning of each of your files. * Explain how your code was integrated to the code of other members of the group. * Marking of this section will also include assessment on the submitted files, the overall functionality of your work, as well as defence of your work during the demonstration. |
| **Only my work is submitted and not the entire working project. Final code from everyone was not uploaded to GitHub in a timely manner and therefore not enough time to link everything together.**  **I personally did the signup/register and login features on the website.**  **All code within the files are entirely my own and not a product of other team member’s code, except the Gov HTML/CSS components.**  **My code is split into two parts – backend and frontend**  **Frontend:**  **Header.js - the page header that is used on all pages confirming it is a Gov UK website**  **Footer.js - the page footer that is used on all pages with the Gov UK Copyright**  **SignUp.js, SignUp.css - contains the code that allows users to create an account with the variables firstName, lastName, zipCode, email, password. It uses a form to do this.**  **SignIn.js, SignIn.css - contains the code that allows users to log in with their email address & password**  **Landing.js - a page with the confirmation panel confirming user has successfully logged in**  **Main.js - checks whether the signup form is complete and that the user has logged in, if so it re-directs the user to the “landing page” which is the page confirming they have logged in**  **App.js - contains the main functionality of the frontend app combining all the components together to make the web page**  **Backend:**  **db.js - checks whether the database (MonoDB) is connected**  **User.js - creates the entry on the database when a user successfully creates an account**  **App.js - contains the main functionality of the backend that allows users to create an account and login**  **https://cloud.mongodb.com/v2/6457e652810cb542aff855cc#/metrics/replicaSet/6457e6be6b18167e78900c44/explorer/doctors/users/find** |

|  |
| --- |
| Your code – Code Quality – Maintainability (10 marks) |
| **Guidance:**   * Comment on the maintainability of your code. * Marking of this section will also include assessment on the submitted files, the overall functionality of your work, as well as defence of your work during the demonstration |
| **Writing clean and efficient code is important for maintainability. I tried my best to make sure that my code was as easy to read as possible. To achieve this I added comments where I thought it was necessary, correctly indented the lines of code to make sure it was easy to read, following naming conventions like using under\_scores or camelCase and being consistent with it (I used camelCase). I also made sure to delete code that I had commented out that I was no longer using (sometimes I would comment code blocks to re-write them).** |

|  |
| --- |
| Your code – Code Quality – version control (10 marks) |
| **Guidance:**   * Comment on your version control and how you ensured that your work was compatible with that of your group members. * Marking of this section will also include assessment on the submitted files, the overall functionality of your work, as well as defence of your work during the demonstration. |
| **There are various pieces of software that are used by developers/software engineers to help them with version control, I.e GitHub, Gitlab, Beanstalk, AWS Code Comit; my team and I said we would use GitHub for version control on this project as most of us were already familiar with it or had some kind of interaction with it previously. It allows you to be working on the same version of the project at all times, and it’s good for seeing how far others have progressed and what they are currently working on, and gives easy access to others’ code if they wanted us to review something or when providing feedback. It’s also important to be able to go back to previous versions of code, especially for identifying and fixing bugs, and working out where things may have gone wrong. Version control is also important for having backups of the code, in case code is deleted or a computer/system fails etc.**  **Unfortunately, the application is not integrated and is not a functioning website, rather just individual components, as people’s finalised code was not uploaded onto GitHub in a timely manner, and therefore there was not enough time to review everything and put the application together successfully, therefore I have only uploaded my own individual work.** |

|  |
| --- |
| Your code – Code Quality – output of test plans (10 marks) |
| **Guidance:**   * List your **test plan and the output** for your individual part of the work and testing the overall application once you integrated with the elements of your group members. * Marking of this section will also include assessment on the submitted files, the overall functionality of your work, as well as deference of your work during the demonstration. |
| |  |  |  |  | | --- | --- | --- | --- | | **Test scenario** | **Test steps** | **Expected outcome** | **Actual outcome** | | **Register an account** | **Enter first name, last name, email, NHS number/postcode, and password, and click submit to make an account** | **Success message** | **Success message – account created** | | **Login with account** | **Enter username & password to log in** | **Success message** | **Success message panel confirming login was successful** | | **Account is saved to backend (on MongoDB)** | **After account creation check the MongoDB** | **All attributes of the account will be listed –first name, last name, Zip code, email address, password** | **All attributes are listed** | |

|  |
| --- |
| Professional conduct – Communication (10 marks)Seeking and using feedback |
| **Guidance:** List here, in a table, the feedback you have received from your team-mates and your tutor and how you used it. |
| |  |  |  |  | | --- | --- | --- | --- | | Feedback | Date received | Source | How you used it | | “Upload your work onto an online repository so that we can do version control” | 22/04/23 | **Kliti Noka** | Uploaded work to GitHub. | | “Don’t hardcode the values as it can make it difficult later if any values need to change” | 22/04/23 | **Kliti Noka** | Made sure that values are able to be easily changed. | | “Try to match the storyboard from coursework 1” | 20/04/23 | **Kliti Noka** | Reviewed storyboard from coursework1 as a reminder. | | “Try to break your code down into smaller reusable components” | 20/04/23 | **Kliti Noka** | Made a separate folder with components and each component now has their own file. | | “Make sure to communicate with the team” | 18/04/23 | **Kliti Noka** | Organised a discord meeting with the group to establish where everyone is at with their work. | |

|  |
| --- |
| Giving constructive feedback |
| **Guidance:** List here, in a table, the CONSTRUCTIVE feedback you have given to your team-mates and what was the problem you were trying to solve.  Evidence of this should also be in the blog. |
| |  |  |  |  | | --- | --- | --- | --- | | Feedback | Date given | Given to | The problem you were trying to solve | | Clean up the code a bit – for example make sure that indentation looks neat | 26/04/23 | **Kliti Noka** | Make sure code is easy to read | | Remember to follow naming conventions on things like variables – be consistent with variable naming e.g underscores vs camelcase  this\_variable  thisVariable | 20/04/23 | **Kliti Noka** | Following coding conventions | | Add descriptive comments where possible | 20/04/23 | **Kliti Noka** | Helps people who did not write the code to understand it easier | | Delete unnecessary code – code you have commented out that you are no longer using can be removed, or stored in a different file if you wish to keep it just in case | 20/04/23 | **Kliti Noka** | Cleans up and removes unnecessary code | | Use whitespace (empty lines) to separate blocks of code where the code looks very condensed | 20/04/23 | **Kliti Noka** | Code is easier on the eye and separates sections of code | |