|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Behavior**  **Our program should include:** | **Input Example**  **When it receives:** | **Output Example**  **It should return:** |
| 0 | See a small description of what the program does |  |  |
| 1 | Form for the user to enter the date of birth and a button to select gender |  |  |
| 2 | The user is to provide their date of birth | DDMMYYYY | Submit button highlighted |
| 3 | The user should choose the gender | Male or Female | Submit button highlighted – Your Akan Name is “xxxxx” |
| 4 | Date validity check - An invalid day should be (d<=0) or (d>31) | 32 | “Enter a valid date” |
| 5 | Month validity check - An invalid month should be (m<= 0) or (m > 12) | 15 | “Enter a valid month” |
| 7 | Have an array of female Akan names for the seven days of the week |  |  |
| 6 | Have an array of male Akan names for the seven days of the week |  |  |
|  | No entry check – Need to provide a time delay and then timeout | No data | “You have not provided any information. Please enter information to proceed!!” |
|  |  |  |  |
|  |  |  |  |
|  | 1. The project uses one or more javascript functions. 2. Appropriate control flows and logical operators are used in the project. 3. The project makes use of one or more arrays in javascript. 4. The project uses a  HTML input to collect user data. 5. The project uses a custom CSS stylesheet that incorporates cascading, box model and floats. 6. The project works as expected, achieving the functionality required. |  |  |
|  |  |  |  |
|  | TECHNICAL REQUIREMENTS   1. The project contains a well-documented README file. This should have:    * Project name.    * Project description.    * Author(s) information.    * Setup instructions - includes any scripts that need to be run if necessary.    * BDD.    * Technologies used.    * Contact information.    * License and Copyright information. 2. The project code is consistently indented and follows the language syntax. 3. The project is submitted in time. It should be submitted before 6:00 pm Friday unless otherwise stated. 4. The project code is pushed to Github and the project is deployed to gh-pages. 5. The project name, description, and URL to live site are provided on the Github repository. 6. Commits are made regularly with relevant and clear commit messages associated with them. 7. The project should be polished in a portfolio-quality state. |  |  |
|  |  |  |  |
|  | **1 pts - A well-documented Readme file on Github**  A project README that includes: - project or program name - author name - description of project - project setup instructions - link to live site on GitHub Pages - copyright and license information |  |  |
|  | **1 pts**  **20 + commits in the project with well detailed commit messages** |  |  |
|  | **1 pts - Project description and URL are provided.** |  |  |
|  | **1 pts - Proper code indentation and syntax are used.**  The project's code is properly indented and follows the language's syntax. This also means that the code is easy to read and follow up due to consistent indentation and use of syntax. |  |  |
|  | **2 pts – The project is well polished in a portfolio quality state.**  The project is well designed, very visually appealing, easy to work through the website, shows a touch of creativity. This also means it goes above the visual aspect with features like clearing form fields after an event and so much more. |  |  |
|  | **1 pts - Full marks**  The project has a small description on the landing page. |  |  |
|  | **3 pts - Full marks**  The user is able to enter their birthday through a form. |  |  |
|  | **1 pts - Full marks**  The user is able to choose their gender |  |  |
|  |  |  |  |
|  | **3 pts – Full marks**  On submission, the application displays the user's Akan name correctly based on their gende |  |  |
|  | **2 pts – Full marks**  If the user does not enter anything in the form or if the user enters an invalid date or month, the application raises an alert message |  |  |
|  |  |  |  |