**Testing / implementation Log**

**100582204**

Website on VM can be found at: <http://127.0.0.1/SourceCode/index.html>

VM domain URI: appdev@ml-lab-f824b4bb-411f-4cf0-ab02-76fe1a8efec5.southeastasia.cloudapp.azure.com

VM Port: 64379

VM Password: appdevvm2002@

|  |  |  |
| --- | --- | --- |
| Date and time work | Amount of time spent | A description of the work done in this session |
| 05/05/2023 | 1 hour | Planning the assignment and setting milestones, as well as getting acquainted with the REST API. Understanding the API is essential for successfully integrating it into the project and ensuring smooth communication between various components. This session was a valuable starting point for the project's implementation and development, giving a clear direction and a deeper understanding of the REST API, which would be critical for successful execution. |
| 06/05/2023 | 1 hour | Development of data visualisation and manipulation. These would serve as a platform for displaying the project's data insights and providing a user-friendly interface for users to interact with the visualisations. Additionally, JavaScript functions were implemented to allow data manipulation. The goal was to create functionality that would allow users to add data, remove data, view raw data, and edit data. To ensure proper operation and address any issues, I tested each functionality separately.  I made significant progress towards the project's goals by devoting time to creating data visualisation and developing data manipulation functionality. These enhancements would improve the project's usability and provide users with useful tools for interacting with and exploring the data. |
| 06/05/2023 | 30 minutes | Discovered that trying to view the entire database at once resulted in crashes and poor performance. To solve this issue, I implemented pagination, which divides data into smaller, more manageable chunks, making it easier to load and display without taxing the system's resources. This was a critical step in optimising the application and ensuring a more seamless user experience, as it enabled users to navigate through database records more precisely, avoiding performance issues and crashes. By devoting time to identifying the problem and implementing the necessary solution. |
| 07/05/2023 | 45 minutes | implementing database raw appending and adding the ability to hide the table when using the "add row" form, and vice versa. The first task was to add the ability to append raw data to the database, which enables users to add new data quickly and easily. The second task was to improve the user interface by incorporating a toggle mechanism. When users choose to add a new row via a form, the table displaying the existing data is hidden, resulting in a cleaner view and less clutter. The work of this session helps the project achieve its goal of providing efficient data management capabilities and an intuitive user experience. |
| 08/05/2023 | 2 hours | This session focused on implementing the edit data feature and pre-populating the edit forms to make their use easier. Users can edit existing data within the project using the edit data functionality, ensuring data accuracy and relevance. However, I encountered delays and challenges during the implementation process. I discovered that I had been referencing incorrect ID values, resulting in discrepancies and problems retrieving the correct data for editing. Despite the obstacles, I persisted and dedicated the necessary time to resolve the problem.  I successfully implemented the edit data functionality and achieved pre-population of the edit forms by referencing the correct ID values. These enhancements improve the project's functionality and make it more user-friendly, resulting in a smooth and convenient data editing process. |
| 09/05/2023 | 1:30 hours | This session focused on implementing the delete rows functionality and making quality of life changes. Users can delete specific rows or records from the project's dataset using the delete rows functionality, which helps to maintain data cleanliness and overall data integrity. Additionally, several changes were made to improve the user experience, such as improving the responsiveness of the user interface, improving error handling, optimising the performance of specific operations, or adding helpful tooltips or instructions to guide users. The session also reviewed the project from the perspective of a user and identified areas where minor but significant improvements could be made. These enhancements contributed to a more robust and user-friendly system, allowing users to manage and interact with data in a more seamless and efficient manner. |
| 10/05/2023 | 30 minutes | During this session, a bug with the edit button was discovered that rendered it useless on all pages except the first. Further investigation revealed that the problem was caused by a missing page number parameter when using the edit button. To fix this, the code was modified and added the necessary page number parameter to the edit button functionality. This change ensures that the edit button works properly regardless of the page the user is currently on. |
| 11/05/2023 + 12/05/2023 | 7 hours | discovered that I have been using the incorrect API endpoint, resulting in suboptimal functionality and integration. To address this problem, I decided to redo the project, this time ensuring proper use of the API endpoint. I spent time re-evaluating the project's requirements and goals and made the necessary changes to ensure that the implementation matched the intended functionality and used the correct API endpoint. I also put in an extra three hours to ensure the redo was completed successfully. This effort ensures that the project adheres to the intended design and runs smoothly in conjunction with the desired APIs. |
| 12/05/2023 | 30 minutes | The d3.js library was used to create a scatter plot visualisation to investigate and represent the relationship between income and age. The scatter plot was created by writing JavaScript code that made use of the d3.js library's functions and methods. The scatter plot was created by parsing the dataset, plotting each data point, assigning income values to the x-axis and age values to the y-axis, and using appropriate styling and labelling to ensure its clarity and readability. This visualisation would aid in data exploration and analysis by providing valuable insights into the relationship between income and age. |
| 16/05/2023 | 2 hours | Three additional graphs were added to the project's visualisation capabilities to provide additional information and understanding of the data. These graphs were designed to provide additional information and understanding of the data, revealing patterns, trends, and relationships. During this session, three different types of graphs were used to provide a variety of visual representations: a bar chart to compare categorical data, a line chart to display trends and changes over time, and a pie chart to show the proportions and percentages of different categories within a dataset. Depending on the project's specific requirements and preferences, appropriate JavaScript libraries or frameworks were used to implement these graphs. By devoting time to adding these three new graphs, the project's visualisation capabilities were improved, and users were able to gain valuable insights, make informed decisions, and effectively communicate data-driven findings. |
| 16/05/2023 | 15 minutes | conducted tests to ensure the functionality and accuracy of the graphs in relation to data changes. I used a systematic approach to conduct the tests, making deliberate changes to the data and closely observing the resulting changes in the graphs. I paid close attention to details like the positions of the data points, the scaling of the axes, and the overall visual representation of the data. This validation process contributes to a trustworthy and reliable data visualisation system, allowing users to make informed decisions based on accurate and up-to-date data. |
| 16/05/2023 | 30 minutes | The project's pagination functionality has been improved by using client-side pagination as an alternative. This method eliminates the need for additional server requests for each page change, resulting in faster and smoother project navigation. During this session, the existing pagination implementation was restructured to align with client-side pagination principles. The code was changed to handle data rendering and display in a paginated manner, ensuring that only the relevant portion of the data was displayed at any given time. User interface elements like page navigation buttons and dropdowns were also used to make it easier for users to interact with the paginated data. Client-side pagination may have limitations in handling large datasets, but it can be an effective solution for achieving a cleaner codebase and an improved user experience in scenarios where the dataset size is manageable. |
| 17/05/2023 | 25 minutes | The ability to refresh the graphs after certain actions was added to the project's functionality. To accomplish this, code was written that would automatically update and refresh the graphs whenever a claim was modified. This procedure entailed identifying specific events, such as claim edits, deletions, or additions, and then triggering the refresh action in response. When a claim is edited, the code detects the changes and updates the corresponding data points or values in the graphs. When a new claim is added, the code incorporates the new data into the appropriate graph representation. This integration ensures that users have a current and reliable visual representation of the data, allowing them to make informed decisions based on the most up-to-date information. |