**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**

**ON-THE-JOB TRAINING**

**DAILY ACTIVITIES**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME: DOMINGO, JOHN KENNETH NICKO M.** | | **INTERSHIP SITE: WEAVERSOFT** | |
| **DATE** | **ACTIVITIES PERFORMED** | **PROBLEMS ENCOUNTERED** | **ACTION TAKEN** |
| February 13, 2024 | * Developed a simple CRUD application using React.js and Node.js, utilizing a JSON file as the database. * Implemented security measures, including React route guards and JWT-based token validation for both the frontend and backend. * Explored Material UI, as it is the primary component library used in the codebase we will handle. * Integrated Material UI components into the CRUD application to ensure a consistent UI/UX. * Added client-side pagination and a search algorithm as required by our mentor. | * None | * N/A |
| February 14, 2024 | * Installed and configured Bitbucket for version control. * Installed and set up FortiClient VPN to connect to the company's network. * Configured SSH (RSA keys, etc.) for secure authentication. * Set up port forwarding for remote access. * Cloned the repository using a virtual machine provided by the company. * Created a new feature branch based on the assigned task. * Modified buttons on the task page and updated the search algorithm by adding policy\_name in the filter. * Created a "View Logs" button and implemented a dialog to display log details. | * Faced challenges while setting up SSH and the development environment since it was the first time configuring it. * Encountered a permission error when attempting to push to Bitbucket due to a missing SSH key. | * Sought guidance from Sir Eldar and Sir Artem for SSH and environment setup. * Researched the SSH key issue and discovered that the key was not loaded in the SSH agent. * Resolved the issue by manually adding the SSH key to the agent. |
| February 17, 2024 | * Implemented field extraction information by adding an info icon next to each field. The icon provides a tooltip explaining the purpose of each field, improving clarity for users. * Updated the label for the validation step, as it was previously treated as an object instead of a more user-friendly format. * Replaced the info icon with a more detailed textual explanation of what needs to be done in that section, enhancing user guidance. * Added a type indicator next to the key name in the policy field to clarify the expected data type. * Enhanced the user profile feature by enabling users to click on the user icon, which opens a dialog displaying their personal information and company details, improving accessibility and user experience. * Created two feature branches and submitted two separate pull requests (PRs) upon completing the tasks. | * Faced challenges in understanding the structure of dynamic components in the codebase due to their complexity and lack of documentation. * Carefully determined appropriate tooltip messages to provide useful information without being overly detailed. * Struggled to define a clear and concise description for Task 1’s extension while maintaining good UI/UX principles. * Encountered difficulties in designing the UI for the user profile dialog, as it was a static component with no prior reference, requiring a thoughtful approach to layout and accessibility. | * Received guidance from Sir Eldar to better understand the structure and functionality of dynamic components. * Used ChatGPT to refine the wording of the tooltip messages, ensuring they were informative yet concise. * Debugged and analyzed the component structure by working backward, identifying how data was handled dynamically. * Incorporated feedback from Sir Amit, who provided insights on improving UI clarity and effectiveness. * Reviewed the output policy explanation provided by Sir Eldar, which helped refine the approach to defining UI elements. * Researched UI inspirations to design the user profile dialog in a way that aligns with the existing interface and enhances usability. |
| February 18, 2024 | * Renamed "DCS" to "Home" as a straightforward update. * Updated the policy wizard feature by:   + Added a dedicated save button for each step while keeping the standard save button for the final step.   + Implemented React Toastify to display a toast notification for saves on intermediate steps.   + Replaced all alert messages with toast errors and used toast success for the final submission while maintaining navigation on the last step. * Modified the task creation behavior to:   + Prevented redirection to a loading page upon creating a task.   + Instead, closed the modal and reloaded the table to reflect the newly added task. * Fixed a bug in the policy page that caused crashes by:   + Used the stack trace to identify an issue with a title formatting function that did not handle null values.   + Adjusted the function to account for null values, preventing crashes when policies were newly created. * Ensured the avatar icon remained visible in the sidebar when it was collapsed. * Updated the user profile modal to a full-page layout, following the requested UI change. * Submitted each of these updates as a separate pull request. | * Started working on an outdated staging branch, which caused merge conflicts when pulling the latest changes. | * Resolved the merge conflicts with guidance from Sir Artem, who advised configuring Git to disable pull rebase. * Successfully merged conflicts for the first time, ensuring the changes were applied without errors. |
| February 19, 2025 | * Watched Clean Code Lesson 1 (1h 48m) and Lesson 2 (1h 6m) by Uncle Bob on YouTube, as advised by Sir Artem. * Lesson 1 Takeaways:   + Understood that clean code is both a social and scientific responsibility.   + Learned that "The only way to go fast is to go well."   + Recognized that good code should make the reader feel "the author of this code cares about me."   + Noted that function names should be verbs, not nouns.   + Applied the extract, extract, extract principle—functions should do one thing well and avoid excessive if-statements.   + Reviewed the Open-Closed Principle—code should be open for extension but closed for modification.   + Learned that functions should not exceed 100 lines—ideally, not even 20 lines. * Lesson 2 Takeaways:   + Understood that comments are not always a good practice.   + Learned that "Writing a comment is a sign of failure."   + Acknowledged that code should be self-explanatory—fewer comments lead to better code.   + Fixed a bug in the task table:     - Previously, when a task failed, nothing happened.     - Updated the system to display a dialog for logs (similar to completed tasks) instead of redirecting.   + Worked on a bug fix merge from yesterday. | * Encountered a merge conflict in the bug fix:   + The file had syntax errors regardless of what was accepted.   + Discovered that Dyze and I had both worked on the same file. | * Compared both versions to merge progress without conflicts while keeping both of our contributions. * Resolved the syntax errors and ensured the bug fix was properly applied. |
| February 20, 2025 | * Watched Clean Code - Uncle Bob / Lesson 3 as assigned by Sir Artem. * Removed all comparison components, including the tab. * Fixed a page crash in the policy page by adding defensive checks. * Made the profile page dynamic by integrating Redux and persisting user data via cookies. * Implemented edit profile functionality to allow username and email updates. * Encountered API issues when sending profile updates due to backend restrictions. * Fixed a bug in the policy page by adding defensive checks in RenderListItem. | * Encountered an edit profile API request failure due to the PUT method not being enabled and an authorization error. * Identified that the policy page bug was caused by unhandled null values in a 1,000+ line file. | * Pushed the profile page updates despite API blockers, as advised by Sir Artem. * Added defensive checks in RenderListItem to handle null values. * Informed the team about backend API issues, which Sir Artem planned to address. |
| February 21, 2025 | * Implemented a three-step registration process. * Developed Step 1 for company name but left validation pending due to the lack of a backend API. * Implemented Step 2 for username and email. * Implemented Step 3 for password and confirm password. * Added validation with input color error indicators, inline error messages, and toast notifications upon submission. | * Could not complete the company name validation feature due to the absence of a backend API. * Discovered a bug in the task page caused by an unnecessary object payload in the request. * Had no additional tasks assigned, as the company did not operate on Fridays. | * Proceeded with implementing the registration steps while keeping the company name validation task on hold until the API became available. * Reported the task page bug to Dyze so he could update the task creation process and resolve the issue. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |