**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** |
| March 10, 2025 | * Started working on the logout API to revoke the token on the server side. * Implemented token blacklisting using SimpleJWT, following the official documentation. * Found out that Jericho had already submitted a pull request for the same task with the same approach. * Did not push my work since the task was already completed. * Began working on the policywizard task to replace the hardcoded URL with the service\_url from the database. * Searched the backend but could not find the service\_url field. * Found tool\_endpoint but was unsure if it was the correct field, so I asked for clarification. * Received confirmation at 10 PM, past my work hours, and decided to continue the task the next day. | * The logout API task was already completed by Jericho, making my work unnecessary. * The service\_url field was unclear, leading to confusion while searching the backend. * Received clarification late, preventing me from finishing the task within working hours. | * Did not push my logout API work since Jericho’s PR already addressed it. * Asked for clarification on the service\_url field before proceeding. * Planned to continue the policywizard task the next day after confirming tool\_endpoint was the correct field. |
| March 11, 2025 | * Completed the assigned task. * Initially created an endpoint to fetch the tool\_endpoint and replaced the hardcoded endpoint with it. * Sir Eldar suggested modifying the approach by making a single endpoint to fetch the tool\_endpoint and handling the request in the server-side class TestFieldViewSet. * Implemented the updated logic in TestFieldViewSet to retrieve the tool\_endpoint dynamically and make requests accordingly. | * Could not test the implementation in the virtual machine due to a lack of data. * The workflow of this feature was not fully explained, making it difficult to verify the expected behavior. | * Notified the superiors that the code was not tested due to missing data. * Waited for their feedback before proceeding with further changes. |
| March 12, 2025 | * nvestigated a bug where the Save button remained disabled after updating the structure of a JSON field in the Policy Wizard. * Initially suspected deep nesting as the cause of the issue. * Used console logs and learned how to use the debugger for deeper inspection. * Tried a deep comparison function but realized through debugging that the compared values were identical. * Worked backward, initially suspecting an issue with state management. * Created an immutable copy of policyData but saw no change. * Discovered that useLocation was updating unexpectedly, causing a rerender. * Traced the rerender to the table component, which assigned policyData = currentPolicy on render. * Fixed the issue by ensuring that unnecessary rerenders did not occur. | * The deep comparison function was not the issue, as debugging showed the compared values were the same. * useLocation was unexpectedly updating, causing unwanted rerenders. * The table component made it difficult to track where the changes were happening. * Found a flaw in the original logic where switching from list to string and back to list cleared the nested array or object. | * Ensured that the component did not rerender unnecessarily. * Reported the issue where switching field types caused data loss. * Fixed the layout issues after Sir Eldar pointed them out. * Received a follow-up task to add an Undo button to prevent data loss and a Close button on the modal. |
| March 13, 2025 | * Updated a pull request to resolve a merge conflict and removed unused code. * Clarified the database content and updated the view to apply the appropriate filter. * Attended a learning session on appropriate HTTP status codes. * Learned that Django’s built-in function unexpectedly returns 404 when no data is found. * Discovered that 204 No Content is also considered inappropriate in some cases, despite being commonly used. | * The explanation of HTTP codes did not fully align with existing documentation, leading to some confusion. | * Resolved the merge conflict and removed unnecessary code from the pull request. * Adjusted the view filtering logic to match the updated database structure. * Engaged in the learning session to improve understanding of HTTP response codes and their correct usage. |
| March 14, 2025 | * Had no assigned tasks in Trello. * Continued reviewing Next.js to enhance skills for future tasks. * Learned more about useMemo for optimizing performance. * Studied useContext for managing global state. * Explored React Query for handling server state efficiently. * Practiced client-side filtering techniques for better UI responsiveness. |  | * Used the available time to deepen knowledge of Next.js and React best practices. * Focused on improving performance optimization and state management techniques. |