Date	# Hours	Description of work
03.02.2025	2	I started by creating my Spring Boot project and setting up the pom.xml file, ensuring the correct dependencies were included for the initial setup. I added a welcome.html page as the main landing page and configured a HomeController to serve it properly. Along the way, I encountered SQL database connection issues, which I resolved by correctly configuring application.properties with the necessary MySQL setting. I implemented SecurityConfig.java, allowing public access to specific pages so that I wouldn't need to log in every time. Additionally, I had issues displaying images, which I resolved by placing them in the static/images/ directory and updating the image paths in the HTML.
04.02.2025	1	I worked on integrating my IntelliJ IDEA project with an existing GitHub repository through the desktop version, ensuring proper Git initialization, remote repository linking, and successfully committing and pushing my code.
05.02.2025	1.5	I worked on refining the user registration process for the login page in my application. After users click "Register" on the login page, they are redirected to a user type selection page. I configured Spring Security to allow access to the /select-user-type endpoint, ensuring that users can view this page without needing to authenticate. Additionally, I updated the SecurityConfig to manage redirections correctly and resolved an HTTP 403 (Forbidden) error by adjusting the necessary permissions. I also verified that the "Go Back" button on the selection page accurately redirects users back to the login page.
07.02.2025	2.5	I worked on setting up the user registration flow for the NativeSpark project. Currently, I am implementing a multi-step registration process where users first select their type: Business, Indigenous Entrepreneur, or Basic User. For business users, I have designed a two-step process. In Step 1, users provide their email, password, and user type, which are stored in the users table. In Step 2, users enter their business details and upload a logo, which is saved in the business_users table. I have set up the necessary entities (User and BusinessUser), as well as repositories, services, and controllers to manage data persistence and form submission. Additionally, I created Thymeleaf templates for the registration pages. I also spent time debugging issues related to database mapping. Despite these efforts, saving BusinessUser data in the database is still not functioning correctly, and I will need to continue troubleshooting this issue in the next session. I also plan to implement a similar process for the other two user types.
09.02.2025	3	I focused on debugging the issue where user data wasn't saved in the database after Step 2 of registration. Initially, there were errors with the @JoinColumn reference in the BusinessUser entity. After fixing this, I encountered another problem: uploaded logo files were not being saved correctly, leading to an error. I updated the BusinessUserService to ensure the upload directory exists before saving files. The application was trying to save files in a non-existent directory within the Tomcat temporary workspace. I modified the code to create the directory if it didn't exist and changed the file-saving logic to use `Files.copy()`, ensuring proper storage of uploaded files.

		I implemented the Entrepreneur User Registration process, following the same structure as the Business User Registration. Similar to the Business Registration, the entrepreneur registration consists of two steps. In Step 1, the user enters their email and password, which are stored in the users table with the user_type set to "ENTREPRENEUR." In Step 2, the user provides personal details, including their first name, last name, a brief description about themselves, indigenous identity, and a profile photo. This information is saved in the entrepreneur_users table and is linked to the corresponding user. Writing logs and progress report 1.

I began by setting up my Spring Boot project, configuring the `pom.xml` file, and ensuring that all necessary dependencies were included. I created a `welcome.html` landing page and implemented a `HomeController` to serve it properly. During the development process, I encountered SQL database connection issues, which I resolved by correctly configuring the `application.properties` file for MySQL. Additionally, I faced challenges with displaying images, which I fixed by placing them in the `static/images/` directory and updating the image paths in the HTML.

To streamline development, I implemented `SecurityConfig.java`, which allows public access to specific pages, thereby avoiding the need to log in repeatedly. I also integrated my IntelliJ IDEA project with GitHub, ensuring proper version control by initializing Git, linking a remote repository, and successfully committing and pushing my code.

A key focus of my work was refining the user registration process for the NativeSpark project. Users can now select their type—Business, Indigenous Entrepreneur, or Basic User—before proceeding with registration. For business users, I designed a two-step process: Step 1 collects their email, password, and user type, storing this information in the `users` table, while Step 2 gathers business details and logo, saving it in the `business_users` table. I set up the necessary entities, repositories, services, and controllers to manage this flow and created the corresponding Thymeleaf templates. Throughout this process, I encountered issues with database mapping and file uploads, which I resolved by ensuring proper reference to the username and modifying `BusinessUserService` to handle file storage correctly.

Next, I implemented the Entrepreneur User Registration process, following the same two-step structure as Business Registration. In Step 1, the user's email and password are stored in the `users` table with the `user_type` set to "ENTREPRENEUR". Step 2 collects the first name, last name, a brief bio, Indigenous identity, and a profile photo, saving this data in the `entrepreneur_users` table. To manage the registration process, I created the `EntrepreneurUser` entity, repository, service, and controller. I also built Thymeleaf templates for entrepreneur registration and ensured proper redirection between the steps. While debugging, I resolved issues related to database persistence and file uploads, similar to those encountered in business registration.

The files/folders I have checked in the repo are as follows:

SecurityConfig.java

BusinessUserController.java

EntrepreneurUserController.java

HomeController.java

LoginController.java

UserRegistrationController.java

User.java

BusinessUser.java

EntrepreneurUser.java

UserRepository.java

BusinessUserRepository.java

EntrepreneurUserRepository.java

UserService.java

BusinessUserService.java

EntrepreneurUserService.java

business-register.html

business-info.html

entrepreneur-register.html

entrepreneur-info.html

select-user-type.html

login.html

welcome.html

ValeriiaN_Progress_Report.xlsx

ValeriiaN_Progress_Report1.docx