Cegeka Homework for Session12 - Agile

Project Ideas :

- CONTROL4 SMART HOMES
- Renting Platform for Camping Equipment's
- Archaeological Sites Monitoring Application

Archaeological Sites Monitoring Application Agile Planning

Short Description: We want to make a monitoring application for archeological sites present in our or other countries to discover and protect our shared human story.

First of all, in our application we will need more databases. The most important are those in which the geographical coordinates of the sites and the one with users will be stored.

EPIC: Creating databases	
USER STORY: As an application developer, I need a database to store the archaeological sites.	**USER STORY:** As an application developer, I need a database to store the users.
TASK: Import all geographic and historical data for sites	**TASK:** Create the logic behind the tables

TASK: Create the logic behind the tables	**TASK:** Create the user and admin tables
TASK: Create the sites tables	

Secondly, we want to create an API in order to prevent data exposes, too many round-trip requests and to provide the right amount of functionality for the consumers.

EPIC: Wrap the Database with a REST API
USER STORY: As an application developer, I need an API that exposes our application data but also ensure API longevity to support future needs.
TASK: Add pagination support to our API for all search.
TASK: Ensure that our API return Proper Response Codes.
TASK: Validate Input Data

Third, we need to develop the user interface and implement it in the backend. We should our async practice between backend/frontend We should make sure that in every sprint we have a "full-stack" team that is able to complete all the work.

EPIC: Designing App UI	
USER STORY: As a user, I need a login page, register page and a visualization page to see all the archeological sites in my area.	**USER STORY:** As a user I need a search bar, a history view for each archeological site and a language selector.
TASK: Create login page	TASK: Create the search bar
TASK: Create register page	TASK: Create the history view
TASK: Create visualization page	**TASK:** Create the language selector

EPIC: Implement the Backend logic	
USER STORY: As an application developer, I need to implement the logic for the login page, register page and the visualization page.	**USER STORY:** As an application developer, I need to implement the logic for the search bar, history view and the language selector.
TASK: Implement the functionality for the login page.	**TASK:** Implement the functionality for the search bar.
TASK: Implement the functionality for the register page.	**TASK:** Implement the functionality for the history view.
TASK: Implement the functionality for the visualization page.	**TASK:** Implement the functionality for the language selector.

Fourthly, the user should have a clean interaction at the first use. That is why we must make sure that we have a well-developed login page interaction.

EPIC: Improve the login page
USER STORY: As a user, I would like the validation on the login page to be very clear so that I can easily see when/if I make a mistake when I log in.
TASK: Implement the pop-ups for the invalidity of the username or password.
TASK: Implement the redirection towards the registration page if the user doesn't have an account.

Sixth, we need to implement security against SQL injection. SQL injections are one of the most utilized web attack vectors used with the goal of retrieving sensitive data from organizations.

EPIC: Implement SQL Injection security
USER STORY: As an application developer, I need to implement the logic to avoid SQL Injection vulnerabilities in my web application.
TASK: Validate user-supplied input for expected data types
TASK: Configure proper error reporting and handling on the web server and in the code so that database error messages are never sent to the client web browser

Finally, we should test our app in order to know if all the components work properly together. We also need to check if the application can be easily scalable in the future. This last epic should be repeated until all bugs are discovered and the application becomes optimal.

EPIC: Testing the Appplication	
USER STORY: As a developer, I need to verify that the user-application interaction is working.	**USER STORY:** As a developer, I need to check the ability of my application to cope with the increase in workload.
TASK: Create tests to validate that everything works fine	**TASK:** Unload the database server from functions it shouldn't run
TASK: Check the real-time response	**TASK:** Verify if the App has continuous availability