IT239 Project 1
Score: ______/150 v1.2

The repository is empty. You will add your own files to it. Only one group member needs to accept the assignment. Students are admins in their own repositories, so you should add your group mates to your repository as a collaborator.

Specifications

This project will simulate a real-world development experience by incorporating client interaction, Agile methodologies, CI/CD pipelines, cloud-based development, and kanban-based project management. You will work in teams of 3–4 people, with each team including at least one Software Development Program (SDP) student and one Information Technology (IT) student. Tasks should be appropriately divided, with IT students focusing on infrastructure-related responsibilities and SDP students handling programming and system integration.

You will create a corporate inventory management system. Your client is a large, global company that manufactures computer micro-components. Since they are global, they want a distributed application, so naturally, you decided to develop this application in the cloud. They want a front-end web application that runs seamlessly and identical in major browsers (Firefox, Chrome, Edge, etc.). No mobile version is required. A user should not be able to view any pages unless they are properly authenticated.

This corporation has several manufacturing plants all over the globe, and your application should track every part at every plant. Users belonging to one plant should only see their parts, but higher level employees (such as executives) should be able to see all parts. For each part, the system should display details such as its part number, display name, specifications, quantity at specific plants, and vendor information. Privileged employees, such as managers, must have the ability to modify the inventory using the web interface.

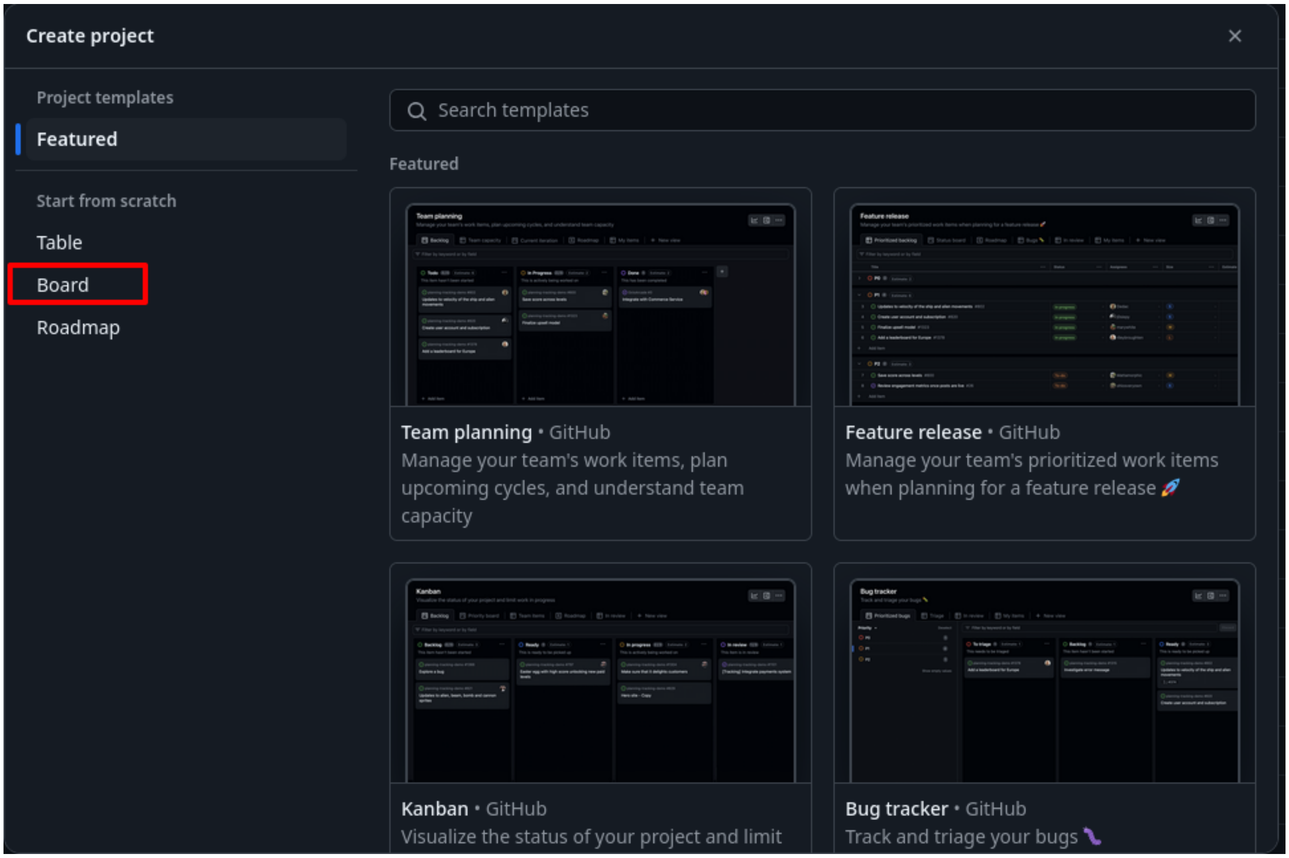
Requirements can change during development, so choose a development model that supports a flexibility. Vague requirements should be cleared up with the client (your instructor). There are no formal checkpoints during the project, so it is your team's responsibility to adapt to changes and ensure the final product meets expectations by the deadline.

Occasionally, the client will request status reports, which will be submitted as Canvas assignments. These reports must be typed, formal, and respond to specific prompts, which may vary by team. Be prepared to provide updates in a professional manner.

Version control is essential for this project. You must use Git to manage your codebase, create a branching strategy, and conduct code reviews through pull requests. No code should be merged into the production branch without a thorough review! Proper use of Git is critical to ensure smooth collaboration within your team.

Constraints

- Your cloud host must be Microsoft Azure.
  - All students should sign up for the GitHub Student Developer Pack. This gives you $100 dollars in Azure credit without requiring a credit card.
  - https://education.github.com/pack
  - You should look at all of the other benefits you get!
- Your database(s) and web application must be hosted on Azure.
- Source code must be hosted on GitHub. Use the assignment link. You can store other things in the repository as well, such as documents.
- CI must be set up and handled by GitHub Actions. A developer should **never** push their code directly to the production environment.
- Authentication must **not** be hard-coded into the application. Authenticating will happen via a username and password via a log in page.
  - If using a database for user/account management, password's must **not** be stored in plaintext!! Proper hashing, salting, and/or peppering should be used.
- There must be server-side and client-side coding. You can use PHP, ASP.NET, or any other server-side framework.
- You must track all of your changes using a kanban board. Use GitHub Projects for this. Your client will occasionally check your board and will be unhappy if it is not being used properly!
  - When setting up your GitHub project, you must use a board, not a table!



Team planning • GitHub
Manage your team's work items, plan upcoming cycles, and understand team capacity

Feature release • GitHub
Manage your team's prioritized work items when planning for a feature release 🚀

Kanban • GitHub
Visualize the status of your project and limit

Bug tracker • GitHub
Track and triage your bugs 🐛

- IT students will be responsible for infrastructure (managing Azure, setting up CI, creating and managing databases/authentication, etc.). SDP students will be responsible for writing code and supporting their IT members as needed.
    - Some cross-training might be a good idea, especially if there are limited IT or SDP students.
- CMS platforms, such as WordPress, cannot be used.
- Your GitHub repository must have a README.md file and a useful .gitignore file.

Submission

Your final submission will be a Git commit ID and a demo to your client. You will be asked to show off infrastructure.

Tips

- Spend some time brainstorming ideas before starting. Develop strategies and a good development methodology. If using scrum, designate a scrum master.
- You will certainly have to use official and 3rd party documentation during this project.
- For the web application, you can use Bootstrap, TypeScript, Blazor, or any other framework/stack.
    - Choose a framework everybody feels comfortable with!
- SDP students, consider using Entity Framework for database mapping.
- Remember to keep your client involved in your project! Ask them questions, for advice, whether or not they like something, etc.

Academic Honesty Policy

Any work submitted by a student that is not their original work will be considered a violation of the school’s academic dishonesty policy. This includes, but is not limited to, using AI to generate code, using code found online, and stealing work from another group.

While collaboration is permitted, sharing work with other groups will be considered a violation of the school’s academic dishonesty policy.

Late Work Policy

Late work will not be accepted for final deliverable. Do not wait to the last hour to submit your project. Make sure your commit is exactly as you want it. If your submission is late, you will receive a 0 for the project.