CEN 3031: Development Environment and Project Proposal Plan

TEAM INFORMATION

- Team Name: Voluntell
- Github Link
- Team Members & Roles:
  - Matthew Strenges (Product Manager)
  - Marc Bishop (Scrum Master)
  - Thomas Yao (Development Team Member)
  - Aiden Zepp (Development Team Member)

PRODUCT INFORMATION & VISION

- Programming Language: C++ & HTML
- What is the product to be developed?
  - A volunteer matching service that matches prospective volunteers with volunteer activities they may be interested in.
- Who are the target customers and users?
  - The target customers are two main groups: individuals looking to volunteer in certain areas or topics of interest, and organizations in need of more volunteers.
- Why should customers buy this product?
  - Our service offers a unique value proposition, primarily targeting the organizations. While the platform is free for prospective volunteers, recognizing their invaluable contribution as a key resource, we present a monetization model for organizations seeking enhanced visibility. By opting for our premium service, these organizations can significantly increase the visibility of their advertisements and listings, among other features. This feature is particularly beneficial for organizations eager to attract new volunteers or those looking to expand their reach to a broader audience. Given the inherent motivation of these organizations to engage with enthusiastic volunteers, investing in our enhanced services becomes a strategic decision. It ensures that their opportunities are prominently displayed, thereby increasing the likelihood of attracting dedicated and aligned volunteers. In summary, our platform not only facilitates a crucial connection between volunteers and organizations but also offers a robust solution for organizations to amplify their presence and impact within the volunteering community.
- Describe your solution to the challenge statement.
  - In today's society, where the desire to contribute positively to social causes is on the rise, many individuals find themselves eager to volunteer but are unsure where to start. Our solution, a volunteer matching service, is designed to bridge this gap by seamlessly connecting prospective volunteers with organizations in need of their passion and skills. By leveraging personalization algorithms, volunteers'

unique interests and skills will be matched with those organizations whose missions and needs resonate, ensuring a meaningful volunteering experience. Ultimately, our solution is not just a matching tool; it's a facilitator of change. It aligns with the increasing consumer trend of supporting impactful initiatives and significantly contributes to tackling issues like poverty, environmental conservation, and community development.

- ❖ Develop a project vision using the Geoffrey Moore Template
  - ➢ For individuals who are eager to volunteer and organizations in need of volunteers,
  - ➢ Who are facing challenges in finding the right match for their volunteer needs or opportunities,
  - ➢ The Volunteer Matching Service developed by Team 10
  - ➢ Is a platform that leverages C++ and HTML to seamlessly connect prospective volunteers with organizations that match their interests and skills,
  - ➢ That provides a unique value proposition by personalizing the volunteering experience and offering organizations a premium service for enhanced visibility,
  - ➢ Unlike traditional volunteering platforms or generic job boards,
  - ➢ Our product utilizes advanced matching algorithms to ensure a meaningful connection between volunteers and organizations, empowering users to contribute to social causes effectively while helping organizations amplify their impact and reach.
  - ➢ By focusing on the mutual benefits for both volunteers and organizations, our service aims to become the go-to platform for impactful volunteering, facilitating change and promoting community engagement at a larger scale.

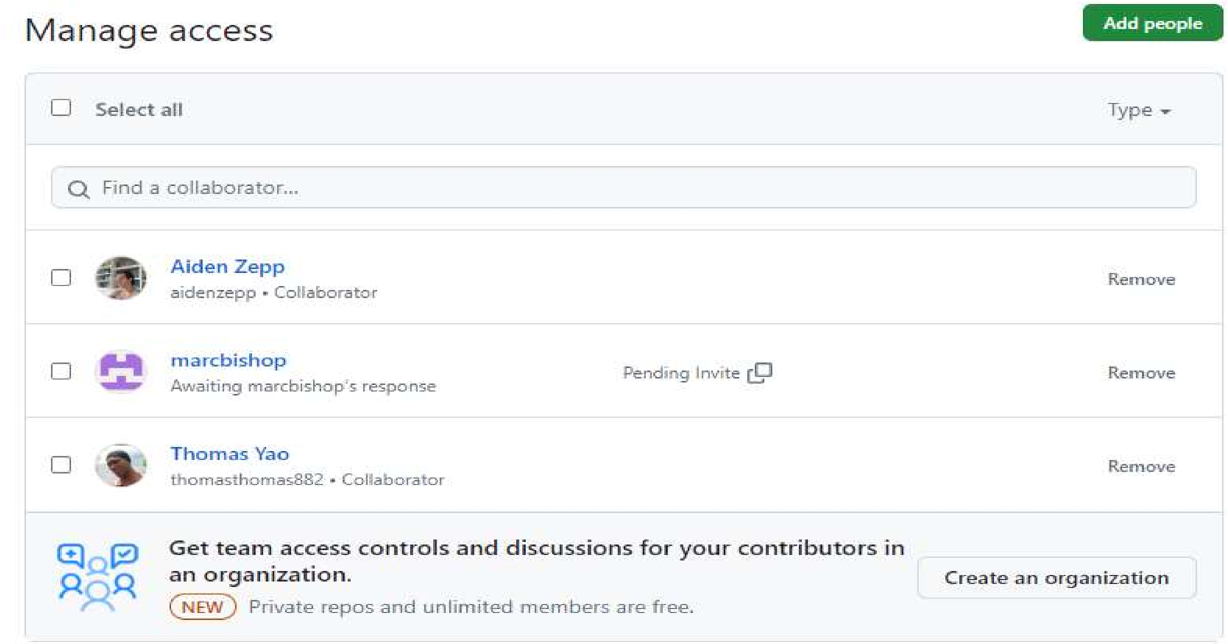
- ❖ Develop a risk management plan using the risk management process.

Explanation	Probability	Effects	Mitigation Strategies
Limited expertise in C++ and HTML among team members	Moderate	Tolerable	Use open-source tools, seek online resources and tutorials
Compatibility issues with browsers and devices	Moderate	Tolerable	Prioritize open-source development tools,

			stay informed about best practices
Short 2-month time frame	High	Serious	Prioritize essential features, use agile development methods, set clear project milestones and deadlines
Small team of 4 members	Low	Serious	Distribute responsibilities effectively, maintain open and efficient communication within the team, focus on core functionalities
Team members falling ill or dropping from the team	Moderate	Serious	Cross-train team members, develop contingency plans, maintain a list of potential replacements
Changes in requirements mid-development	Moderate	Serious	Establish a clear change management process, document changes, and adapt to evolving requirements
Absence of users for the project	High	Serious	Focus on developing a proof of concept or prototype, consider future potential users, and adapt for broader use cases

Underestimating project completion time	Moderate	Serious	Regularly review project progress, adjust timelines if necessary, and internally track project milestones

GITHUB SETUP



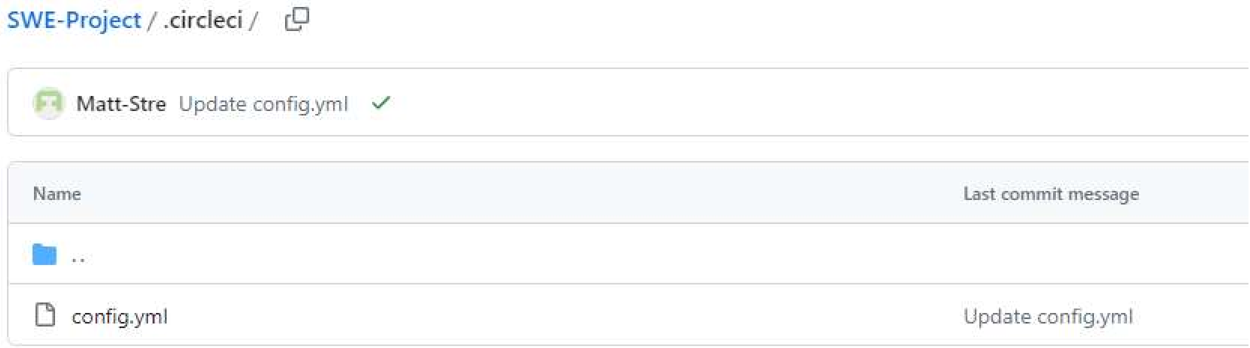
Manage access

Aiden Zepp
aidenzepp • Collaborator

marcbishop
Awaiting marcbishop's response — Pending Invite

Thomas Yao
thomasthomas882 • Collaborator

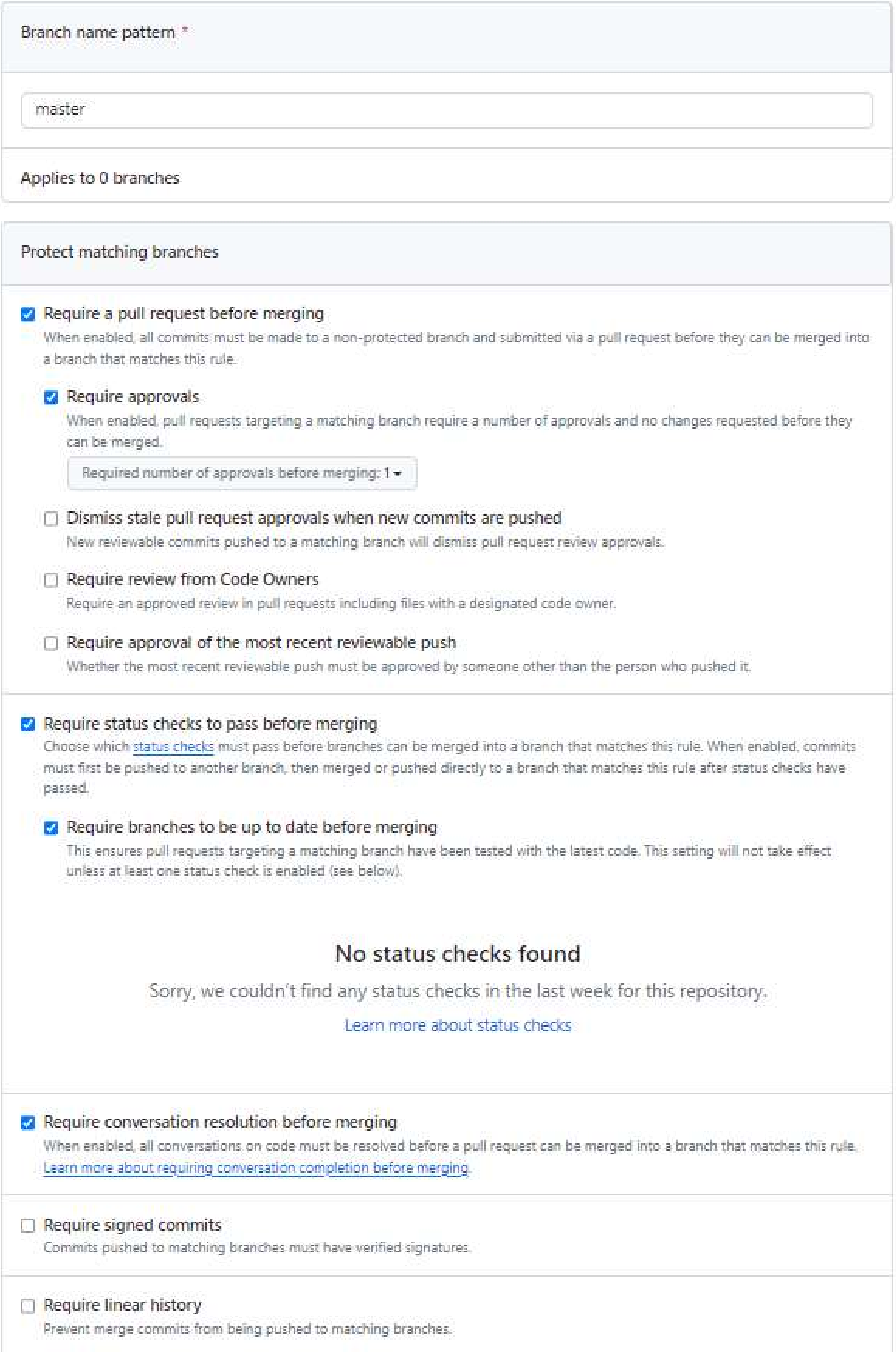
Get team access controls and discussions for your contributors in an organization.
NEW Private repos and unlimited members are free.



SWE-Project / .circleci /

Matt-Stre Update config.yml ✓

Name	Last commit message
..	
config.yml	Update config.yml



Branch name pattern *

master

Applies to 0 branches

Protect matching branches

☑ Require a pull request before merging
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

☑ Require approvals
When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.

Required number of approvals before merging: 1 ▾

□ Dismiss stale pull request approvals when new commits are pushed
New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

□ Require review from Code Owners
Require an approved review in pull requests including files with a designated code owner.

□ Require approval of the most recent reviewable push
Whether the most recent reviewable push must be approved by someone other than the person who pushed it.

☑ Require status checks to pass before merging
Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

☑ Require branches to be up to date before merging
This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

No status checks found

Sorry, we couldn't find any status checks in the last week for this repository.

Learn more about status checks

☑ Require conversation resolution before merging
When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule. Learn more about requiring conversation completion before merging.

□ Require signed commits
Commits pushed to matching branches must have verified signatures.

□ Require linear history
Prevent merge commits from being pushed to matching branches.