Security Test for Threat Analysis

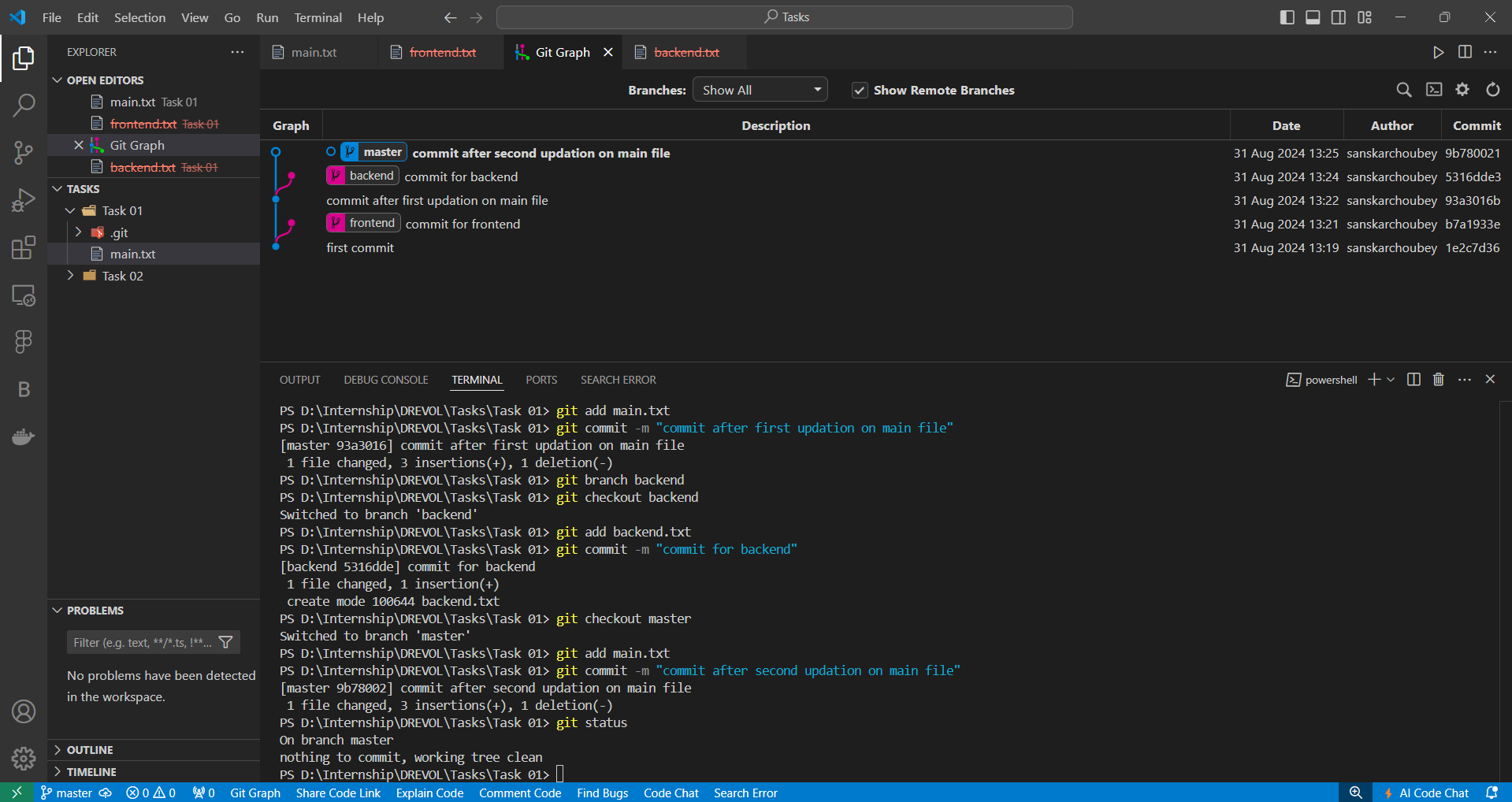
# Objective

The primary goal of this task is to set up a security test environment to analyze potential threats. This involves setting up a secure code repository, allowing controlled access, defining safety thresholds, and establishing a notification system in case of security breaches.

# Steps Involved

## 1. Set of Files/Code Branches

Objective: Create a structured and secure repository to manage the files and code branches for the project.  
Tools/Technologies: Git, GitHub/GitLab/Bitbucket.  
Steps:  
- Initialize a Git repository to manage version control.  
- Create multiple branches for different features or modules (e.g., main, development, feature-xyz).

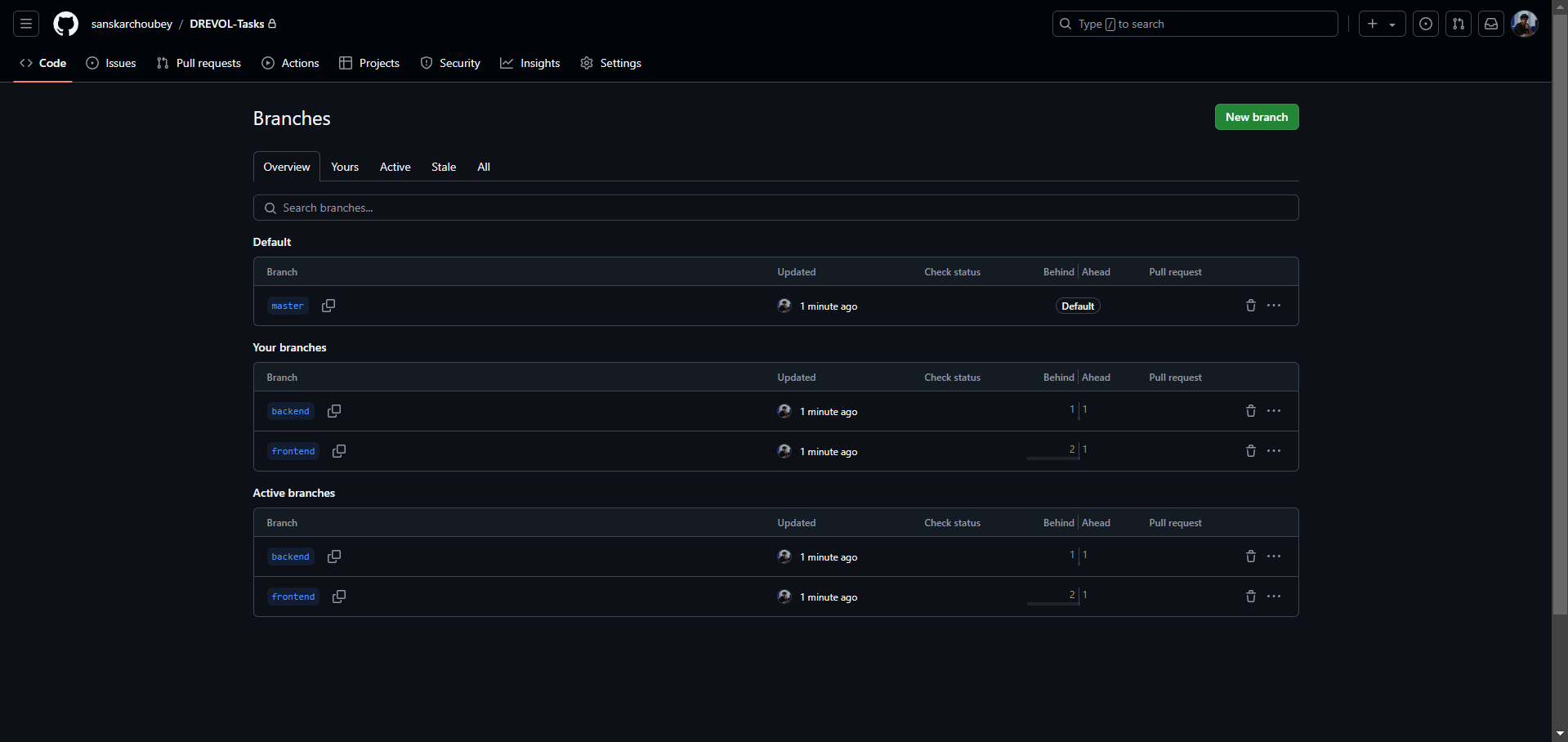
  
- Ensure that branch naming conventions and branch protection rules are established.

## 2. Store the Files

Objective: Securely store the files and code branches in a centralized repository.  
Tools/Technologies: GitHub/GitLab/Bitbucket (private repositories).  
Steps:  
- Push the local repository to a remote server (e.g., GitHub).  
- Ensure the repository is private and only accessible to authorized users.

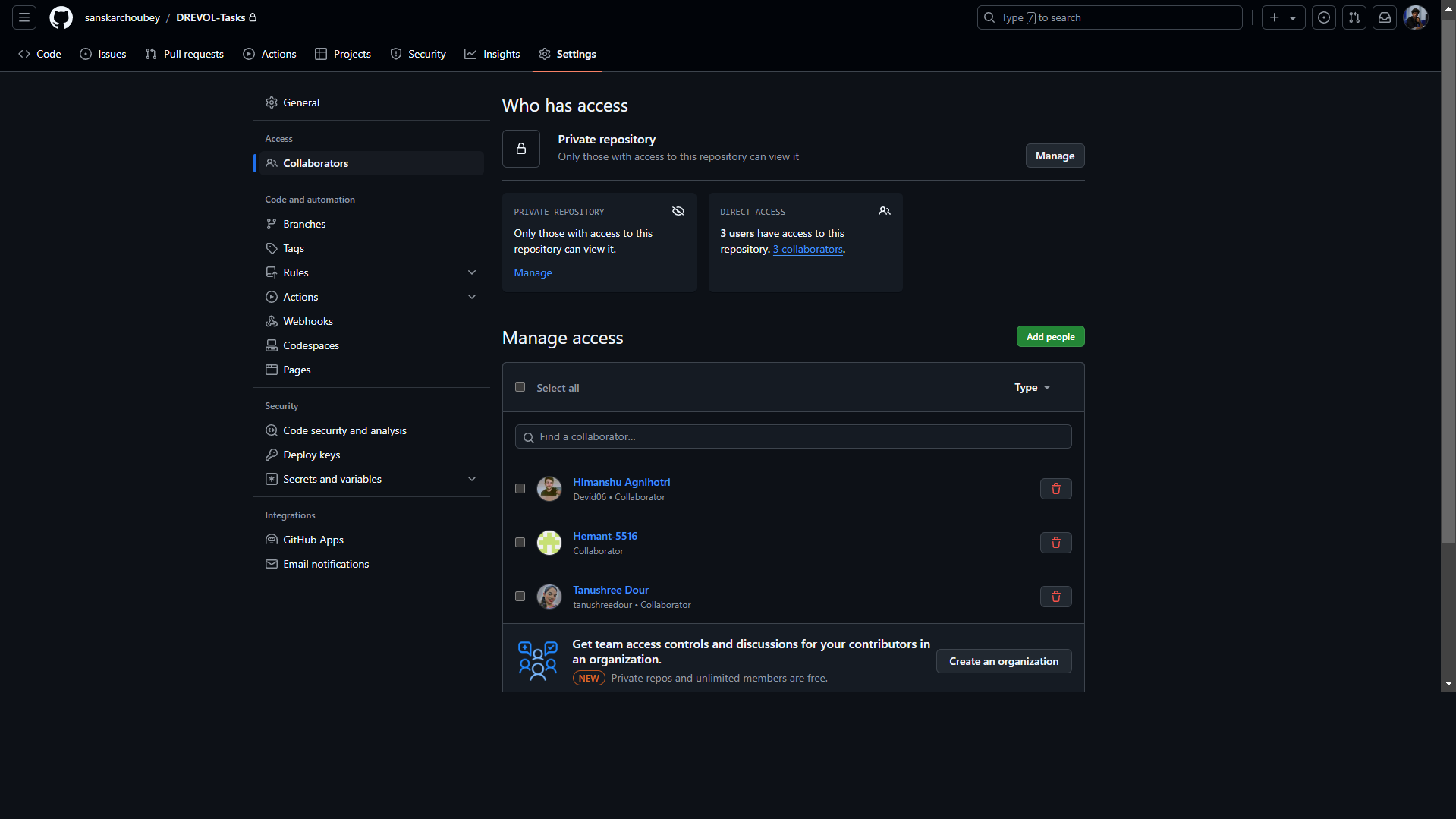
A screen shot of a computer

Description automatically generated



## 3. Secure Access for 3 Programmers

Objective: Provide secure access to three programmers while ensuring the repository's integrity.  
Tools/Technologies: GitHub/GitLab/Bitbucket, SSH keys, 2FA.  
Steps:  
- Invite the 3 programmers to the repository with appropriate roles (e.g., Collaborator).

  
- Implement SSH keys and/or 2FA to enhance security.  
- Define access levels (read/write permissions) based on the roles of each programmer.

## 4. Define Safety Thresholds

Objective: Establish thresholds for detecting unauthorized access or suspicious activities.  
Tools/Technologies: SonarQube (for code quality), Snyk (for dependency scanning).  
Steps:  
- Set up File Integrity Monitoring (FIM) to detect unauthorized changes to files.  
- Implement code quality checks using tools like SonarQube to identify vulnerabilities.  
- Conduct dependency scans to ensure that third-party libraries are secure.  
- Define specific thresholds for alerts (e.g., unauthorized branch changes, code quality drops).

## 5. Notification System for Threat Detection

Objective: Create an automated notification system to alert the team in case of a security breach.  
Tools/Technologies: GitHub Actions/GitLab CI, Slack, Email.  
Steps:  
- Set up automated workflows using CI/CD tools (e.g., GitHub Actions) to monitor the repository.  
- Configure alerts for any security thresholds being breached.  
- Set up notifications to be sent via Email, or other communication channels to the relevant team members.

A screenshot of a computer program

Description automatically generated