INFO1111: Computing 1A Professionalism

2025 Semester 1

Skills: Team Project Report

Submission number: T4 SL Group 6 (1)

Github link: `https://github.com/KaltsitFan/INFO1111_GROUP.git`

Team Members:

Name	Student ID	Target * Foundation	Target * Advanced	Selected Major
Fan Kaffa	510041359	AG	NA	Computer Science
Lin Link	540801400	A	NA	Data Science
Song Jared	550155230	A	NA	SW Development
Zheng Barnett	550718806	A	NA	Cyber Security

* Use the following codes:

- NA = Not attempting in this submission
- A = Attempting (not previously attempting)
- AW = Attempting (achieved weak in a previous submission)
- AG = Attempting (achieved good in a previous submission)
- S = Already achieved strong in a previous submission

Contents

1. Group Response

Our team, composed of members from different computing majors, effectively utilized GitHub's issue tracker for task allocation, enabling clear responsibility assignments and progress tracking. Cross-disciplinary collaboration played a crucial role—our Software Development member assisted the Cybersecurity member in coding challenges, while our Data Science member guided data-related tasks. Initially, LaTeX documentation presented significant challenges due to frequent syntax errors and lack of real-time previews. Transitioning to Visual Studio Code with LaTeX extensions resolved these issues by enabling immediate error detection and correction. Through structured GitHub management and mutual technical support, our collaborative approach markedly improved communication, efficiency, and overall project management effectiveness.

2. Individual Response

2.1. Skills for Computer Science: Kaffa Fan

In the disaster response system project, my primary responsibilities included the establishment of the overall system architecture, the setup of the GitHub collaboration framework, and the configuration of the LaTeX documentation environment. Reflecting based on the SFIA framework, I selected the following two skills:

Software Development (PROG)

According to the SFIA framework [1], software development is foundational to implementing technological solutions. For the LA wildfire disaster response scenario, effectively managing documentation using LaTeX was critical to ensure rapid generation and dissemination of accurate reports. Initially, our team frequently encountered LaTeX syntax errors such as missing commands missing commands (\enditemize) due to using editors without real-time preview capabilities. By transitioning to Visual Studio Code with LaTeX extensions, we could immediately identify and rectify these errors, significantly reducing documentation delays. This directly supported emergency response requirements, enabling quick delivery of clear instructions and updates to responders. Additionally, my demonstrated ability to compile and manage documents via the Git terminal further ensured smooth, efficient project documentation workflows.

Skill Enhancement (Team Collaboration Details): Introducing real-time preview tools considerably enhanced the team's productivity and accuracy, minimizing errors and speeding up response times—key factors during an emergency.

Areas for Further Improvement: Despite improvements, I identified an over-reliance on tool-based error detection. In future projects, I aim to improve my manual code review skills, ensuring reliability even when technological resources are limited, a critical capability in disaster situations.

Systems Integration and Build (SINT)

According to the SFIA framework [1], effective systems integration is essential for creating a coherent and responsive disaster management system. In the context of the wildfire scenario, my role involved integrating various system components, such as real-time data interfaces and a user-friendly incident dashboard for emergency personnel. Early in the project, unclear task assignments created confusion among team members. Implementing GitHub Issues significantly clarified roles and responsibilities, specifically tasks like "Frontend Interface (Incident Dashboard)" and "Data Interface Development (Real-time Sensor Feeds)," streamlining integration processes. My consistent use of Git for task management and module synchronization (Figures 4) ensured efficient module integration, crucial for rapid deployment in emergency scenarios.

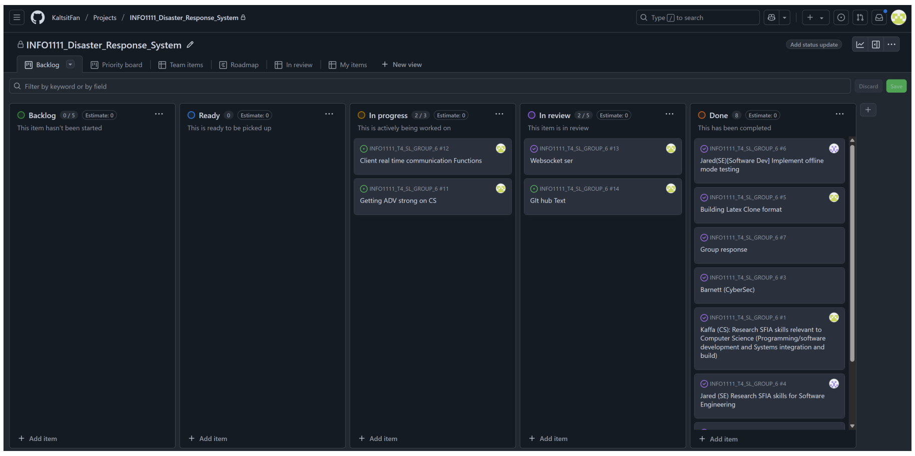
Skill Enhancement (Team Collaboration Details): Clear task definition through GitHub substantially improved our integration capability, facilitating quick and smooth development of crucial disaster-response features, such as real-time dashboards.

Areas for Further Improvement: Our technical discussions sometimes involved excessive jargon, complicating interdisciplinary communication. Moving forward, simplifying complex technical concepts into clear, accessible language will be essential, particularly in high-stakes, collaborative disaster-response environments.

2.2. Advanced Submission Kaffa (Computer Science)

Project Management and Progress Overview

Throughout the project, I established and utilized GitHub Projects along with Issue tracking functionalities to coordinate tasks, assign responsibilities, and monitor overall project progress effectively. The built-in project chart feature provided by GitHub enabled team members to visually track task completion rates, facilitating a clear understanding of the project's progression and enhancing our overall productivity.



KaltsitFan / Projects / INFO1111_Disaster_Response_System
Type / to search
INFO1111_Disaster_Response_System
Add status update
Backlog
Priority board
Team items
Roadmap
In review
My items
New view
Filter by keyword or by field
Discard
Save

Backlog 0 / 5 Estimate: 0
This item hasn't been started

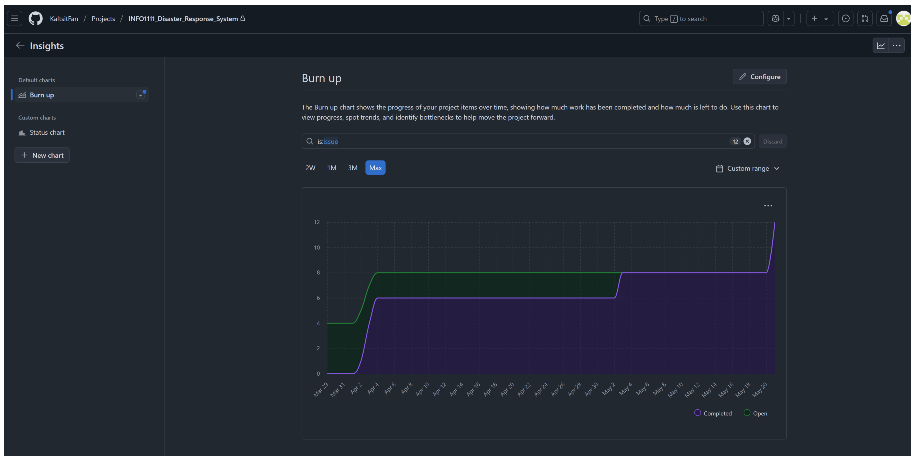
Ready 0 Estimate: 0
This is ready to be picked up

In progress 2 / 3 Estimate: 0
This is actively being worked on
- INFO1111_T4_SL_GROUP_6 #12 Client real time communication Functions
- INFO1111_T4_SL_GROUP_6 #11 Getting ADV strong on CS

In review 2 / 5 Estimate: 0
This item is in review
- INFO1111_T4_SL_GROUP_6 #13 Websocket ser
- INFO1111_T4_SL_GROUP_6 #14 GIt hub Text

Done 8 Estimate: 0
This has been completed
- INFO1111_T4_SL_GROUP_6 #6 Jared(SE)[Software Dev] Implement offline mode testing
- INFO1111_T4_SL_GROUP_6 #5 Building Latex Clone format
- INFO1111_T4_SL_GROUP_6 #7 Group response
- INFO1111_T4_SL_GROUP_6 #3 Barnett (CyberSec)
- INFO1111_T4_SL_GROUP_6 #1 Kaffa (CS): Research SFIA skills relevant to Computer Science (Programming/software development and Systems integration and build)
- INFO1111_T4_SL_GROUP_6 #4 Jared (SE) Research SFIA skills for Software Engineering

Figure 1: GitHub project board displaying Issues and their statuses



Burn up

The Burn up chart shows the progress of your project items over time, showing how much work has been completed and how much is left to do. Use this chart to view progress, spot trends, and identify bottlenecks to help move the project forward.

is:issue 12

Figure 2: GitHub project progress line chart

Tool Exploration: Python WebSockets (Part A)

Main Functionalities of Python WebSockets Python WebSockets offers robust support for real-time bidirectional communication between clients and servers. Its primary functionalities include:

- **Full Duplex Communication:** Allows simultaneous sending and receiving of data.
- **Event-Driven Data Push:** Enables servers to proactively push updates to clients, eliminating continuous polling and thus ideal for real-time applications.

- **Seamless Integration and Support:** Python's websockets library integrates effortlessly with the asyncio framework, simplifying asynchronous programming and client-server interactions.

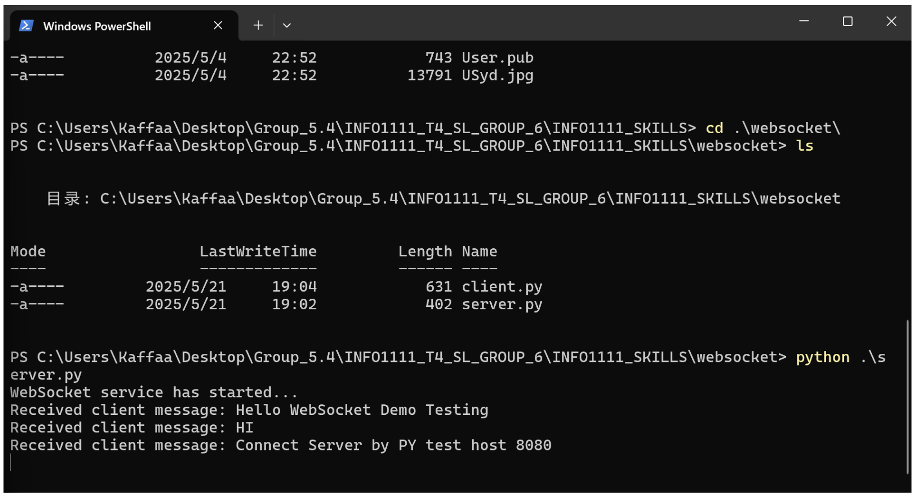
Importance in Computer Science WebSockets play a pivotal role in applications that require real-time interaction, such as online collaboration tools, multiplayer games, and real-time monitoring dashboards. Specifically, in disaster response scenarios, WebSockets significantly improve system responsiveness by providing instantaneous data updates, enabling emergency responders to quickly react and mitigate risks efficiently.

Limitations of WebSockets Despite its significant advantages, WebSockets has certain limitations:

- **Limited Efficiency for Media Streams:** WebSockets is less efficient for streaming audio and video.
- **Complexity in Managing Connection Reliability:** Developers need to manually handle reconnection logic after interruptions, increasing implementation complexity.
- **Resource Intensive:** Continuous connections consume substantial server resources, especially in large-scale systems, requiring careful infrastructure planning and optimization.

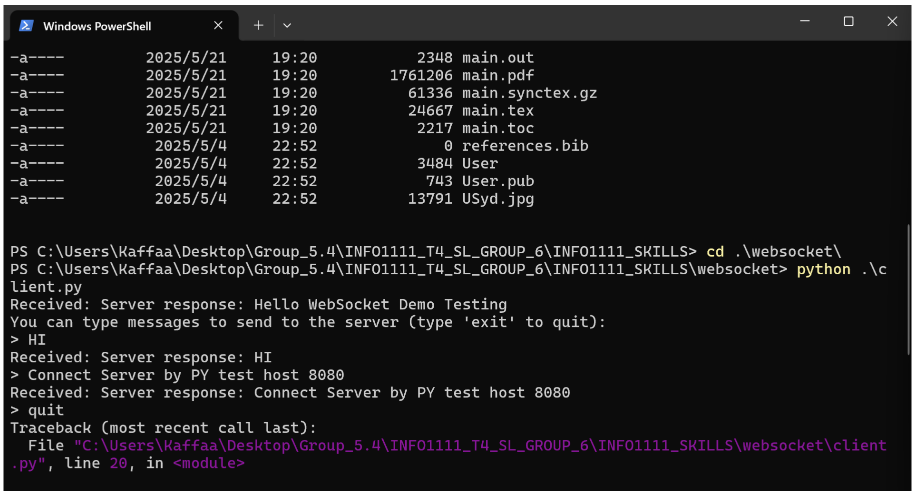
Technical Implementation and Reflective Learning (Part B)

Technical Implementation: Basic WebSockets Example I developed a basic WebSockets demonstration showcasing client-server interaction. The application successfully establishes a WebSocket connection enabling real-time messaging, effectively illustrating core WebSocket functionality and typical application scenarios.



```
-a----          2025/5/4     22:52            743 User.pub
-a----          2025/5/4     22:52          13791 USyd.jpg


PS C:\Users\Kaffaa\Desktop\Group_5.4\INFO1111_T4_SL_GROUP_6\INFO1111_SKILLS> cd .\websocket\
PS C:\Users\Kaffaa\Desktop\Group_5.4\INFO1111_T4_SL_GROUP_6\INFO1111_SKILLS\websocket> ls


    目录: C:\Users\Kaffaa\Desktop\Group_5.4\INFO1111_T4_SL_GROUP_6\INFO1111_SKILLS\websocket


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2025/5/21     19:04            631 client.py
-a----         2025/5/21     19:02            402 server.py


PS C:\Users\Kaffaa\Desktop\Group_5.4\INFO1111_T4_SL_GROUP_6\INFO1111_SKILLS\websocket> python .\s
erver.py
WebSocket service has started...
Received client message: Hello WebSocket Demo Testing
Received client message: HI
Received client message: Connect Server by PY test host 8080
```

Figure 3: Terminal output demonstrating server startup and client connections



```
-a----         2025/5/21     19:20           2348 main.out
-a----         2025/5/21     19:20        1761206 main.pdf
-a----         2025/5/21     19:20          61336 main.synctex.gz
-a----         2025/5/21     19:20          24667 main.tex
-a----         2025/5/21     19:20           2217 main.toc
-a----          2025/5/4     22:52              0 references.bib
-a----          2025/5/4     22:52           3484 User
-a----          2025/5/4     22:52            743 User.pub
-a----          2025/5/4     22:52          13791 USyd.jpg


PS C:\Users\Kaffaa\Desktop\Group_5.4\INFO1111_T4_SL_GROUP_6\INFO1111_SKILLS> cd .\websocket\
PS C:\Users\Kaffaa\Desktop\Group_5.4\INFO1111_T4_SL_GROUP_6\INFO1111_SKILLS\websocket> python .\c
lient.py
Received: Server response: Hello WebSocket Demo Testing
You can type messages to send to the server (type 'exit' to quit):
> HI
Received: Server response: HI
> Connect Server by PY test host 8080
Received: Server response: Connect Server by PY test host 8080
> quit
Traceback (most recent call last):
  File "C:\Users\Kaffaa\Desktop\Group_5.4\INFO1111_T4_SL_GROUP_6\INFO1111_SKILLS\websocket\client
.py", line 20, in <module>
```

Figure 4: Example of message exchanges between client and server

Reflective Learning Initially, I encountered challenges related to environment setup and mastering asynchronous data handling, particularly regarding issues such as port mismatches and conflicts causing connection failures. To resolve these problems, I actively engaged with online forums like Stack Overflow, consulted official documentation, and conducted iterative testing with my team. Through this practical exercise, I not only gained deeper insights into WebSockets technology but also understood its strategic value in building highly responsive and real-time systems, especially applicable to critical scenarios like disaster management [2] [3].

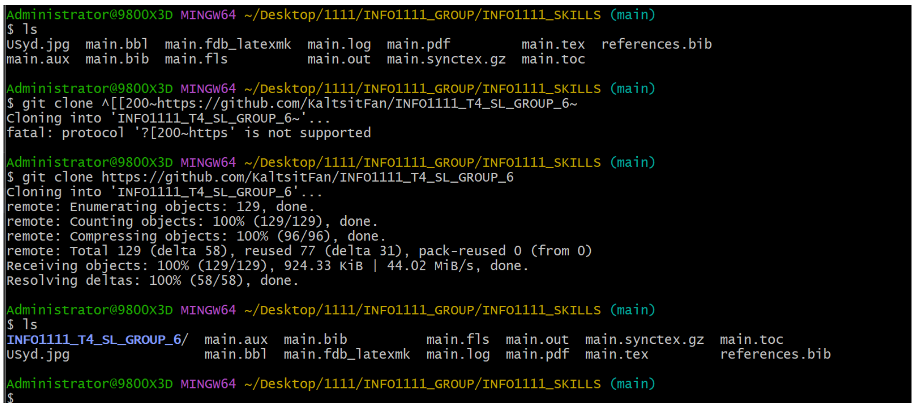
3. Git Response

1. Clone the repository

Clone the remote repository to your local computer:

```
git clone https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6
```

This picture illustrates the cloning process:



```
Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ ls
USyd.jpg  main.bbl  main.fdb_latexmk  main.log  main.pdf         main.tex  references.bib
main.aux  main.bib  main.fls          main.out  main.synctex.gz  main.toc

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ git clone ^[[200~https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6~
Cloning into 'INFO1111_T4_SL_GROUP_6~'...
fatal: protocol '?[200~https' is not supported

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ git clone https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6
Cloning into 'INFO1111_T4_SL_GROUP_6'...
remote: Enumerating objects: 129, done.
remote: Counting objects: 100% (129/129), done.
remote: Compressing objects: 100% (96/96), done.
remote: Total 129 (delta 58), reused 77 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (129/129), 924.33 KiB | 44.02 MiB/s, done.
Resolving deltas: 100% (58/58), done.

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ ls
INFO1111_T4_SL_GROUP_6/  main.aux  main.bib          main.fls  main.out  main.synctex.gz  main.toc
USyd.jpg                 main.bbl  main.fdb_latexmk  main.log  main.pdf  main.tex         references.bib

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$
```

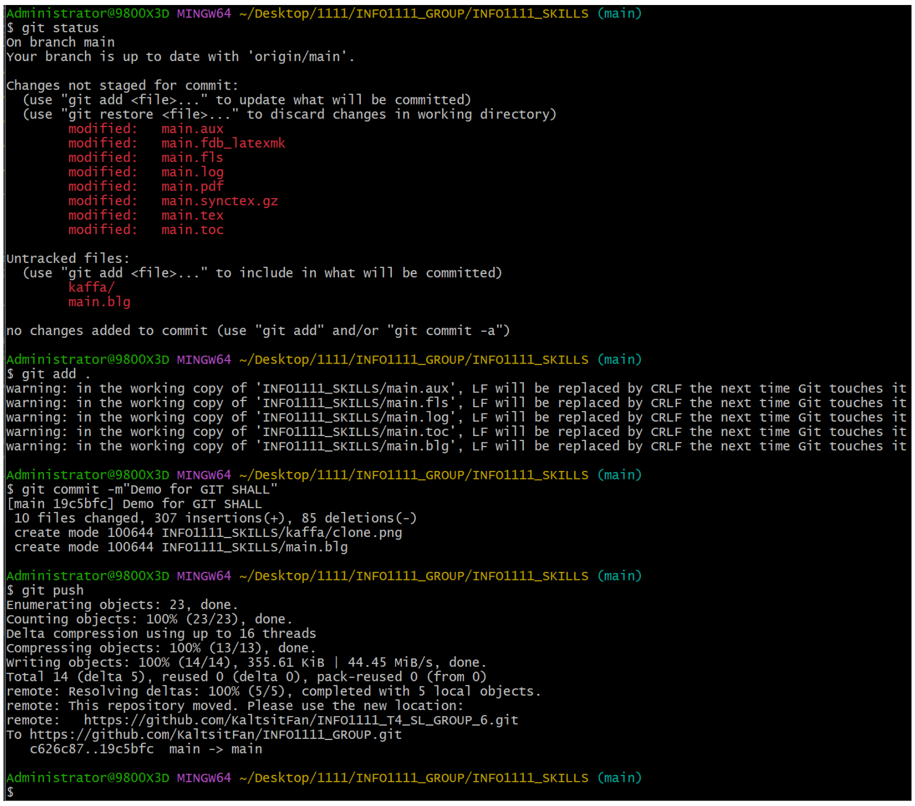
Figure 5: Cloning the repository

2. Stage, Commit, Push, and Check Status

Stage all modified files, commit the changes with a clear message, push the committed changes to GitHub, and verify the repository status:

```
git add .
git commit -m "Your commit message"
git push
git status
```

This picture demonstrates staging, committing, pushing, and checking status:



```
Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.aux
        modified:   main.fdb_latexmk
        modified:   main.fls
        modified:   main.log
        modified:   main.pdf
        modified:   main.synctex.gz
        modified:   main.tex
        modified:   main.toc

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        kaffa/
        main.blg

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ git add .
warning: in the working copy of 'INFO1111_SKILLS/main.aux', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INFO1111_SKILLS/main.fls', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INFO1111_SKILLS/main.log', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INFO1111_SKILLS/main.toc', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INFO1111_SKILLS/main.blg', LF will be replaced by CRLF the next time Git touches it

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ git commit -m"Demo for GIT SHALL"
[main 19c5bfc] Demo for GIT SHALL
 10 files changed, 307 insertions(+), 85 deletions(-)
 create mode 100644 INFO1111_SKILLS/kaffa/clone.png
 create mode 100644 INFO1111_SKILLS/main.blg

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$ git push
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 16 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (14/14), 355.61 KiB | 44.45 MiB/s, done.
Total 14 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
remote: This repository moved. Please use the new location:
remote:   https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6.git
To https://github.com/KaltsitFan/INFO1111_GROUP.git
   c626c87..19c5bfc  main -> main

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (main)
$
```

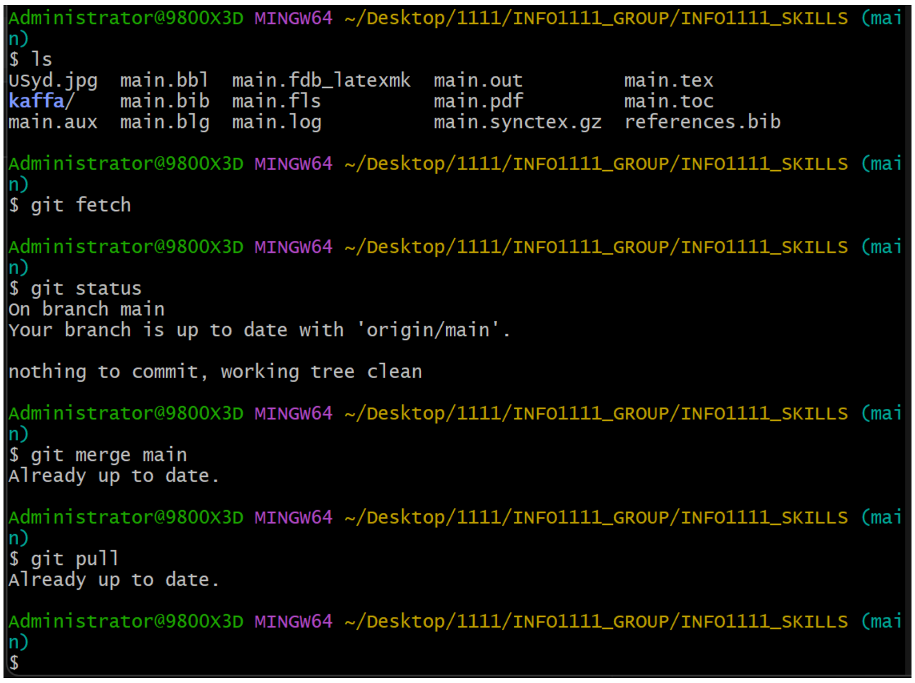
Figure 6: Staging, committing, pushing changes, and checking status

3. Synchronize with remote repository

Pull the latest updates from the GitHub repository to synchronize your local repository:

```
git pull origin main
```

This picture shows synchronizing the local repository with remote updates:



```
Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (mai
n)
$ ls
USyd.jpg  main.bbl  main.fdb_latexmk  main.out         main.tex
kaffa/    main.bib  main.fls          main.pdf         main.toc
main.aux  main.blg  main.log          main.synctex.gz  references.bib

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (mai
n)
$ git fetch

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (mai
n)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (mai
n)
$ git merge main
Already up to date.

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (mai
n)
$ git pull
Already up to date.

Administrator@9800X3D MINGW64 ~/Desktop/1111/INFO1111_GROUP/INFO1111_SKILLS (mai
n)
$
```

Figure 7: Pulling latest changes from remote repository

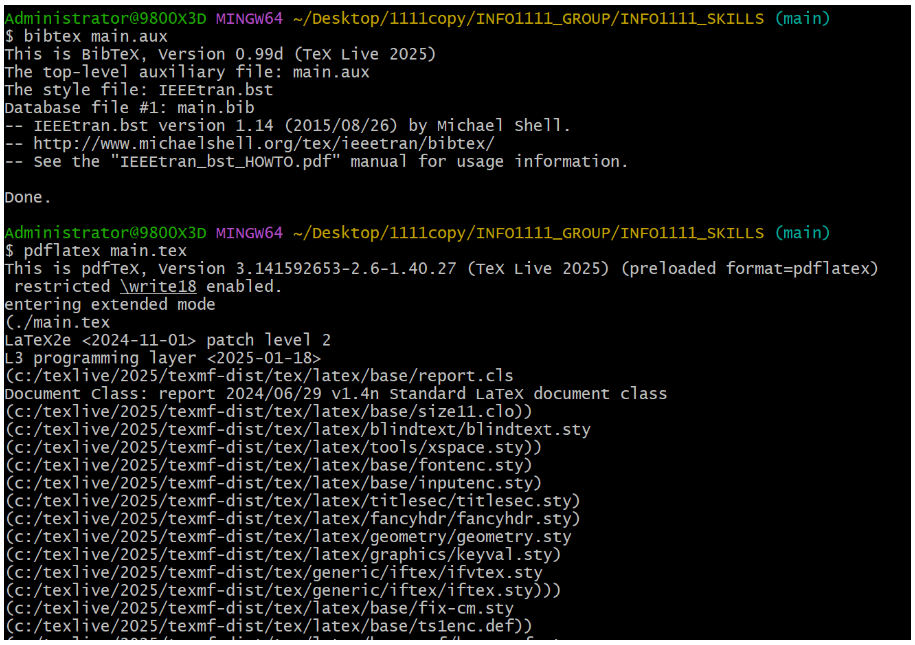
4. Generate PDF using Git Terminal

Compile your LaTeX document into PDF using Git terminal commands:

```
bibtex main.aux and pdflatex yourfile.tex
```

This generates a PDF document from your LaTeX file directly via terminal. Bibtex make sure citation correct

This picture illustrates generating a PDF using terminal:



```
Administrator@9800X3D MINGW64 ~/Desktop/1111copy/INFO1111_GROUP/INFO1111_SKILLS (main)
$ bibtex main.aux
This is BibTeX, Version 0.99d (TeX Live 2025)
The top-level auxiliary file: main.aux
The style file: IEEEtran.bst
Database file #1: main.bib
-- IEEEtran.bst version 1.14 (2015/08/26) by Michael Shell.
-- http://www.michaelshell.org/tex/ieeetran/bibtex/
-- See the "IEEEtran_bst_HOWTO.pdf" manual for usage information.

Done.

Administrator@9800X3D MINGW64 ~/Desktop/1111copy/INFO1111_GROUP/INFO1111_SKILLS (main)
$ pdflatex main.tex
This is pdfTeX, Version 3.141592653-2.6-1.40.27 (TeX Live 2025) (preloaded format=pdflatex)
 restricted \write18 enabled.
entering extended mode
(./main.tex
LaTeX2e <2024-11-01> patch level 2
L3 programming layer <2025-01-18>
(c:/texlive/2025/texmf-dist/tex/latex/base/report.cls
Document Class: report 2024/06/29 v1.4n Standard LaTeX document class
(c:/texlive/2025/texmf-dist/tex/latex/base/size11.clo))
(c:/texlive/2025/texmf-dist/tex/latex/blindtext/blindtext.sty
(c:/texlive/2025/texmf-dist/tex/latex/tools/xspace.sty))
(c:/texlive/2025/texmf-dist/tex/latex/base/fontenc.sty)
(c:/texlive/2025/texmf-dist/tex/latex/base/inputenc.sty)
(c:/texlive/2025/texmf-dist/tex/latex/titlesec/titlesec.sty)
(c:/texlive/2025/texmf-dist/tex/latex/fancyhdr/fancyhdr.sty)
(c:/texlive/2025/texmf-dist/tex/latex/geometry/geometry.sty
(c:/texlive/2025/texmf-dist/tex/latex/graphics/keyval.sty)
(c:/texlive/2025/texmf-dist/tex/generic/iftex/ifvtex.sty
(c:/texlive/2025/texmf-dist/tex/generic/iftex/iftex.sty)))
(c:/texlive/2025/texmf-dist/tex/latex/base/fix-cm.sty
(c:/texlive/2025/texmf-dist/tex/latex/base/ts1enc.def))
```

2.3. Skills for SW Development: Jared Song

Through this project, I identified two critical skills from the SFIA framework relevant to software development:

Key Technical Skills

- **PROG (Programming/Software Development)**
  According to the SFIA framework [1], developing the disaster system's offline functionality required:
  - Implementing local data caching using Python's `shelve` module
  - Writing thread-safe code for concurrent access during emergencies
- **TEST (Software Testing)**
  Establish comprehensive test coverage for disaster scenarios123:
  - Parameterized test suite covering distinct failure modes:
    * Network partitions (simulated with `pytest-timeout`)
    * Data corruption (CRC32 validation tests)
    * Resource exhaustion (memory/stress tests)
  - Mock service framework featuring:
    * Configurable failure injection
    * Latency simulation
    * Stateful behavior modeling

Skill Development through Collaboration

According to the SFIA framework [1], The team environment enhanced these skills by:

- **Cross-domain feedback**: Data Science members' statistical analysis helped refine our cache invalidation algorithm, the data collected has also simplified the work and made the work more straightforward.
- **Collective problem-solving**: Pair programming sessions fixed race conditions in the resource allocator module
- **Tool knowledge sharing**: Learned GitHub Actions CI configuration from Computer Science teammate, which really helps me enhance my skills and understanding of GitHub.

Areas for Improvement

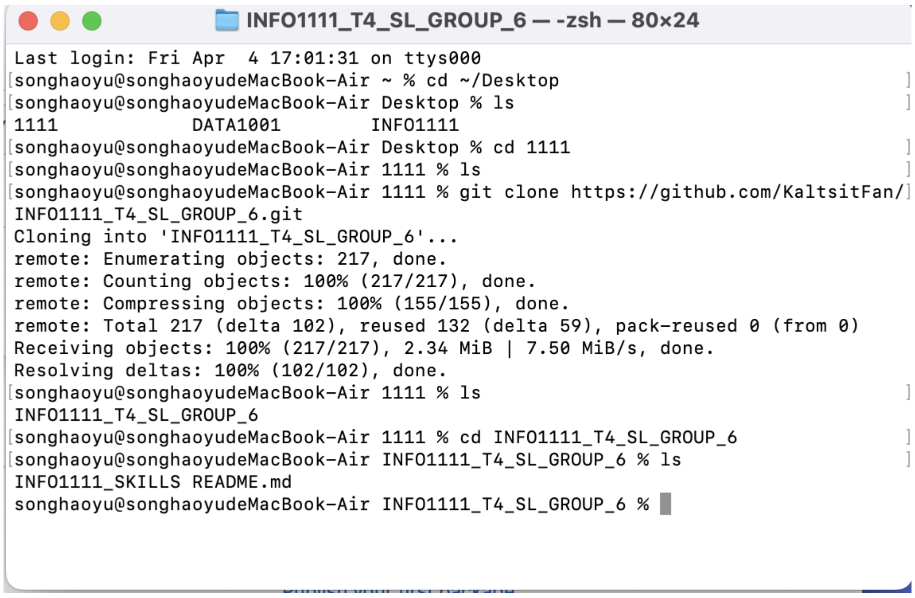
Through my work on the disaster response system, I've identified several technical and professional skills that require refinement:

- **Performance Optimization**: Need deeper understanding of profiling tools (e.g. cProfile) - evidenced when our stress tests failed at 10,000+ concurrent users, be able to learn more about Python and be proficient in using Python.
- **Technical Documentation**: Find difficulty in using Github and Latex, so learning more skills and implementing automated documentation generation are important.

Git Response

1. Clone the repository

Clone the remote repository to local computer:

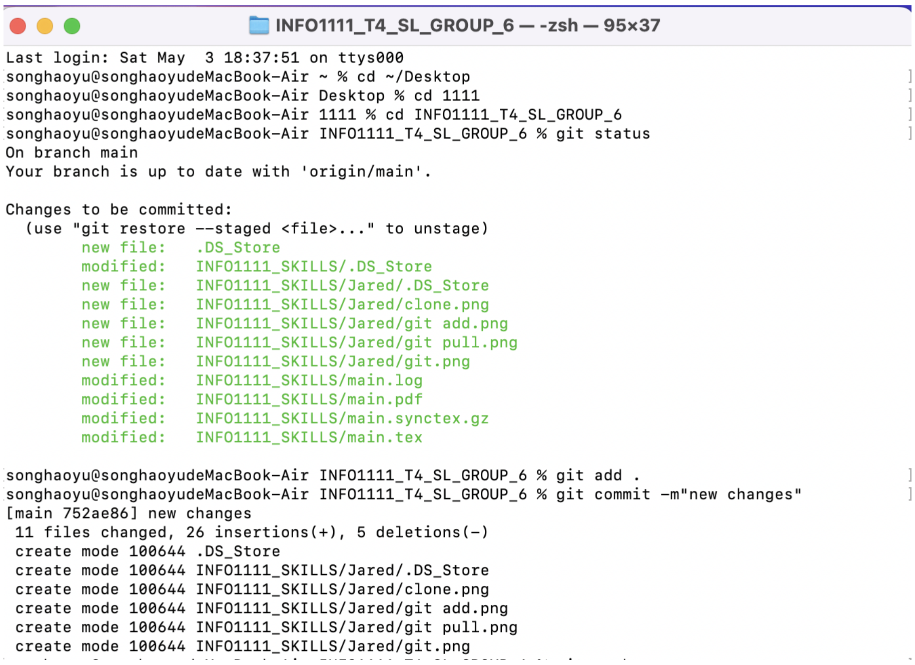


INFO1111_T4_SL_GROUP_6 — -zsh — 80×24

```
Last login: Fri Apr  4 17:01:31 on ttys000
songhaoyu@songhaoyudeMacBook-Air ~ % cd ~/Desktop
songhaoyu@songhaoyudeMacBook-Air Desktop % ls
1111            DATA1001        INFO1111
songhaoyu@songhaoyudeMacBook-Air Desktop % cd 1111
songhaoyu@songhaoyudeMacBook-Air 1111 % ls
songhaoyu@songhaoyudeMacBook-Air 1111 % git clone https://github.com/KaltsitFan/
INFO1111_T4_SL_GROUP_6.git
Cloning into 'INFO1111_T4_SL_GROUP_6'...
remote: Enumerating objects: 217, done.
remote: Counting objects: 100% (217/217), done.
remote: Compressing objects: 100% (155/155), done.
remote: Total 217 (delta 102), reused 132 (delta 59), pack-reused 0 (from 0)
Receiving objects: 100% (217/217), 2.34 MiB | 7.50 MiB/s, done.
Resolving deltas: 100% (102/102), done.
songhaoyu@songhaoyudeMacBook-Air 1111 % ls
INFO1111_T4_SL_GROUP_6
songhaoyu@songhaoyudeMacBook-Air 1111 % cd INFO1111_T4_SL_GROUP_6
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % ls
INFO1111_SKILLS README.md
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 %
```

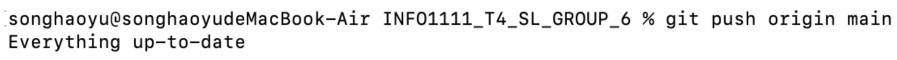
Figure 8: Cloning the repository

2. Commit and push changes to local repository



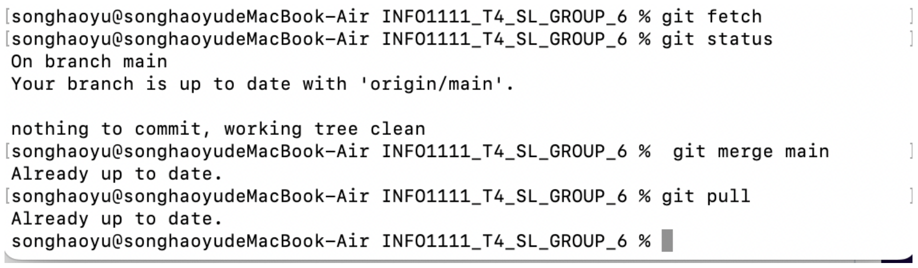
INFO1111_T4_SL_GROUP_6 — -zsh — 95×37

```
Last login: Sat May  3 18:37:51 on ttys000
songhaoyu@songhaoyudeMacBook-Air ~ % cd ~/Desktop
songhaoyu@songhaoyudeMacBook-Air Desktop % cd 1111
songhaoyu@songhaoyudeMacBook-Air 1111 % cd INFO1111_T4_SL_GROUP_6
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .DS_Store
        modified:   INFO1111_SKILLS/.DS_Store
        new file:   INFO1111_SKILLS/Jared/.DS_Store
        new file:   INFO1111_SKILLS/Jared/clone.png
        new file:   INFO1111_SKILLS/Jared/git add.png
        new file:   INFO1111_SKILLS/Jared/git pull.png
        new file:   INFO1111_SKILLS/Jared/git.png
        modified:   INFO1111_SKILLS/main.log
        modified:   INFO1111_SKILLS/main.pdf
        modified:   INFO1111_SKILLS/main.synctex.gz
        modified:   INFO1111_SKILLS/main.tex

songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git add .
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git commit -m"new changes"
[main 752ae86] new changes
 11 files changed, 26 insertions(+), 5 deletions(-)
 create mode 100644 .DS_Store
 create mode 100644 INFO1111_SKILLS/Jared/.DS_Store
 create mode 100644 INFO1111_SKILLS/Jared/clone.png
 create mode 100644 INFO1111_SKILLS/Jared/git add.png
 create mode 100644 INFO1111_SKILLS/Jared/git pull.png
 create mode 100644 INFO1111_SKILLS/Jared/git.png
```



```
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git push origin main
Everything up-to-date
```

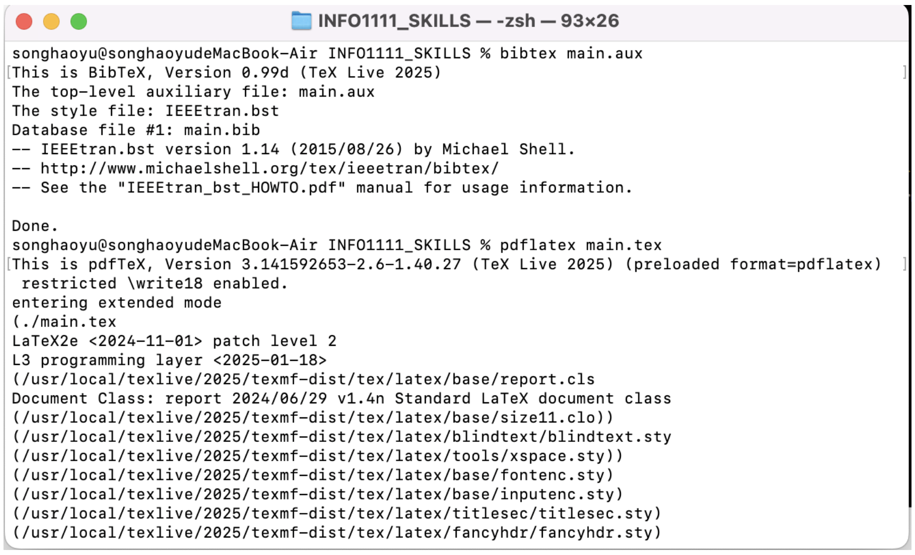
3. Synchronize repository



```
[songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git fetch
[songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
[songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 %  git merge main
Already up to date.
[songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git pull
Already up to date.
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 %
```

Figure 9: Pulling latest changes from remote repository

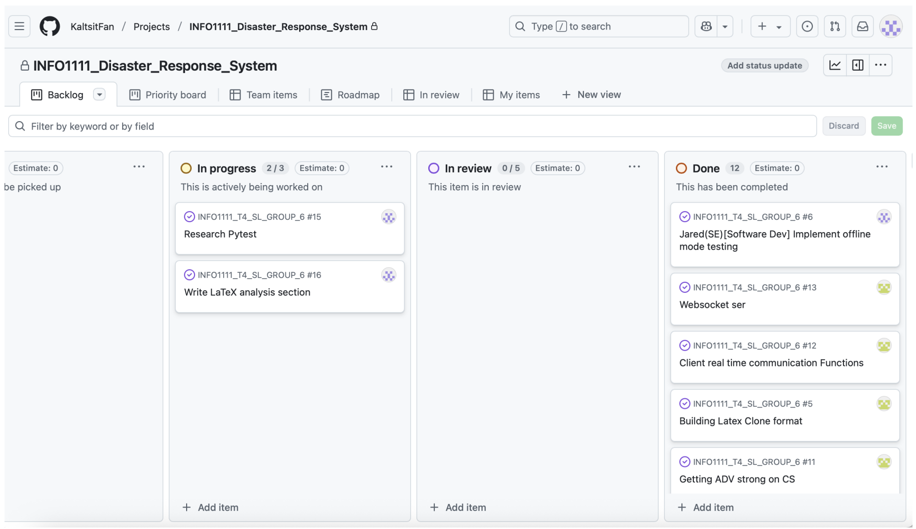
4. PDF Generation Through Git CLI



```
INFO1111_SKILLS — -zsh — 93×26
songhaoyu@songhaoyudeMacBook-Air INFO1111_SKILLS % bibtex main.aux
[This is BibTeX, Version 0.99d (TeX Live 2025)
The top-level auxiliary file: main.aux
The style file: IEEEtran.bst
Database file #1: main.bib
-- IEEEtran.bst version 1.14 (2015/08/26) by Michael Shell.
-- http://www.michaelshell.org/tex/ieeetran/bibtex/
-- See the "IEEEtran_bst_HOWTO.pdf" manual for usage information.

Done.
songhaoyu@songhaoyudeMacBook-Air INFO1111_SKILLS % pdflatex main.tex
[This is pdfTeX, Version 3.141592653-2.6-1.40.27 (TeX Live 2025) (preloaded format=pdflatex)
 restricted \write18 enabled.
entering extended mode
(./main.tex
LaTeX2e <2024-11-01> patch level 2
L3 programming layer <2025-01-18>
(/usr/local/texlive/2025/texmf-dist/tex/latex/base/report.cls
Document Class: report 2024/06/29 v1.4n Standard LaTeX document class
(/usr/local/texlive/2025/texmf-dist/tex/latex/base/size11.clo))
(/usr/local/texlive/2025/texmf-dist/tex/latex/blindtext/blindtext.sty
(/usr/local/texlive/2025/texmf-dist/tex/latex/tools/xspace.sty))
(/usr/local/texlive/2025/texmf-dist/tex/latex/base/fontenc.sty)
(/usr/local/texlive/2025/texmf-dist/tex/latex/base/inputenc.sty)
(/usr/local/texlive/2025/texmf-dist/tex/latex/titlesec/titlesec.sty)
(/usr/local/texlive/2025/texmf-dist/tex/latex/fancyhdr/fancyhdr.sty)
```

2.4. Advanced Submission Jared(Software Engineering) GitHub Project Tracking Evidence

To manage the progress of Advanced Task 2, our team created a GitHub Project board titled *INFO1111_Disaster_Response_System*. Tasks such as researching Pytest, writing LaTeX sections, and collecting screenshots were assigned via GitHub



KaltsitFan / Projects / INFO1111_Disaster_Response_System
Type / to search
INFO1111_Disaster_Response_System
Add status update
Backlog
Priority board
Team items
Roadmap
In review
My items
New view
Filter by keyword or by field
Discard
Save

Estimate: 0
be picked up

In progress 2 / 3 Estimate: 0
This is actively being worked on
- INFO1111_T4_SL_GROUP_6 #15 Research Pytest
- INFO1111_T4_SL_GROUP_6 #16 Write LaTeX analysis section

In review 0 / 5 Estimate: 0
This item is in review

Done 12 Estimate: 0
This has been completed
- INFO1111_T4_SL_GROUP_6 #6 Jared(SE)[Software Dev] Implement offline mode testing
- INFO1111_T4_SL_GROUP_6 #13 Websocket ser
- INFO1111_T4_SL_GROUP_6 #12 Client real time communication Functions
- INFO1111_T4_SL_GROUP_6 #5 Building Latex Clone format
- INFO1111_T4_SL_GROUP_6 #11 Getting ADV strong on CS

+ Add item

Figure 10: INFO1111 GitHub Project board tracking advanced task progress

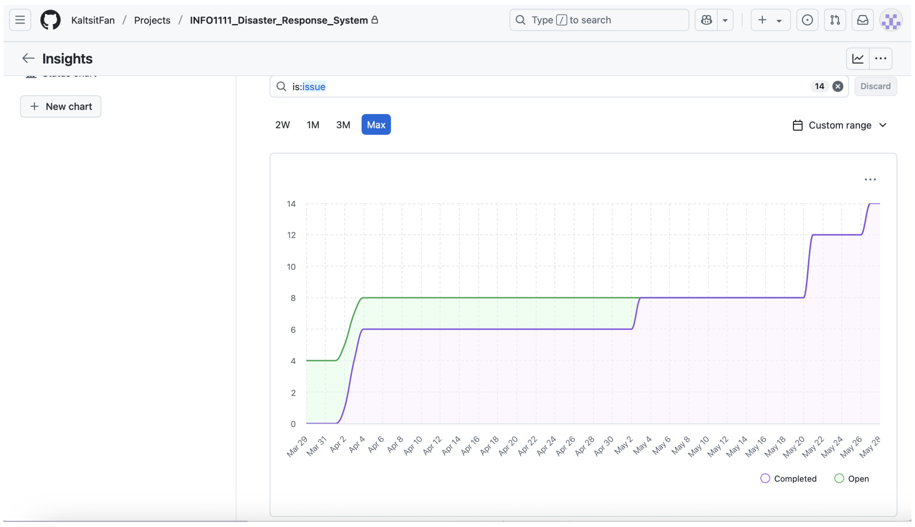


Figure 11: GitHub project chart showing activity over time

Part A: Exploration of Tech Tools

Pytest is one of the most widely adopted testing frameworks in Python-based software development. It is especially valued for its simplicity, scalability, and integration with continuous integration (CI) pipelines [4].

Main Functionalities:

- **Automatic test discovery**: Pytest automatically identifies files and functions that start with `test_` and executes them, which simplifies the testing process.
- **Fixture support**: Pytest provides a powerful fixture system that allows setup and teardown logic to be reused across multiple test functions.
- **Detailed failure reports**: When a test fails, Pytest generates clear and detailed error messages with variable values, making debugging much more efficient.

Relevance to Software Development:

In modern software engineering, testing is essential for ensuring correctness and robustness. Pytest plays a vital role in automating tests, enabling developers to detect regressions and bugs early. It integrates seamlessly with CI tools and GitHub workflows, improving code quality and developer productivity [5]. Within the context of a disaster response system, the reliability of modules such as resource allocation or emergency communication must be assured, and Pytest facilitates this by enabling thorough testing.

Limitations:

- **Steep learning curve for advanced features**: Although basic usage is easy, features like fixture scopes and parameterisation can be difficult for beginners.
- **Limited support for asynchronous code**: Out of the box, Pytest does not support asynchronous functions. Plugins like `pytest-asyncio` are required.
- **Not suited for GUI testing**: Pytest is primarily designed for backend and logic tests; it lacks direct support for GUI elements.

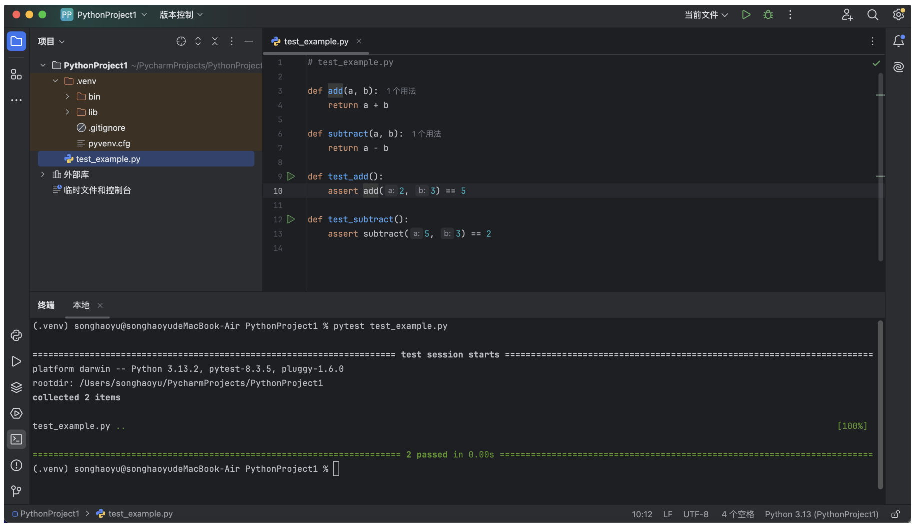
References:

[5], [4], [6], [3]

Part B: Technical Skills and Analysis

To explore the usage of Pytest, I developed a simple example involving a function that adds two numbers. The corresponding test function ensures the correctness of the result.

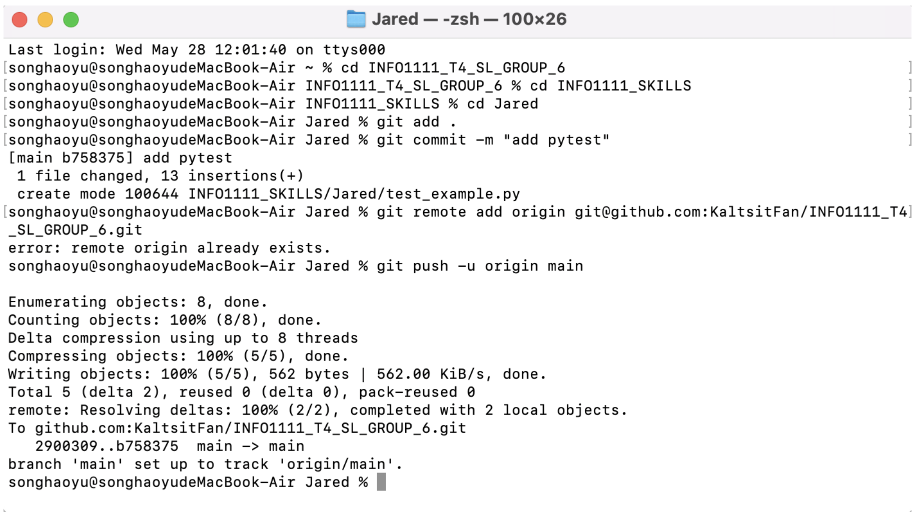
Screenshot Evidence:


PythonProject1
版本控制
当前文件
项目
PythonProject1 ~/PycharmProjects/PythonProject
.venv
bin
lib
.gitignore
pyvenv.cfg
test_example.py
外部库
临时文件和控制台
终端
本地
PythonProject1 > test_example.py
10:12 LF UTF-8 4 个空格 Python 3.13 (PythonProject1)

```python
# test_example.py

def add(a, b):  1个用法
    return a + b

def subtract(a, b):  1个用法
    return a - b

def test_add():
    assert add( a: 2,  b: 3) == 5

def test_subtract():
    assert subtract( a: 5,  b: 3) == 2
```

```
(.venv) songhaoyu@songhaoyudeMacBook-Air PythonProject1 % pytest test_example.py

======================================================================= test session starts ========================================================================
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.6.0
rootdir: /Users/songhaoyu/PycharmProjects/PythonProject1
collected 2 items

test_example.py ..                                                                                                                                            [100%]

======================================================================== 2 passed in 0.00s =========================================================================
(.venv) songhaoyu@songhaoyudeMacBook-Air PythonProject1 %
```

Figure 12: Test function and Pytest result in terminal



```
Last login: Wed May 28 12:01:40 on ttys000
songhaoyu@songhaoyudeMacBook-Air ~ % cd INFO1111_T4_SL_GROUP_6
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % cd INFO1111_SKILLS
songhaoyu@songhaoyudeMacBook-Air INFO1111_SKILLS % cd Jared
songhaoyu@songhaoyudeMacBook-Air Jared % git add .
songhaoyu@songhaoyudeMacBook-Air Jared % git commit -m "add pytest"
[main b758375] add pytest
 1 file changed, 13 insertions(+)
 create mode 100644 INFO1111_SKILLS/Jared/test_example.py
songhaoyu@songhaoyudeMacBook-Air Jared % git remote add origin git@github.com:KaltsitFan/INFO1111_T4
_SL_GROUP_6.git
error: remote origin already exists.
songhaoyu@songhaoyudeMacBook-Air Jared % git push -u origin main

Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 562 bytes | 562.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:KaltsitFan/INFO1111_T4_SL_GROUP_6.git
   2900309..b758375  main -> main
branch 'main' set up to track 'origin/main'.
songhaoyu@songhaoyudeMacBook-Air Jared %
```

Figure 13: Git commit and push of Pytest test file

Reflection:

I began my learning process by reading the official Pytest documentation [4] and watching short tutorials online. Initially, I found the fixture system difficult to understand, especially regarding different scopes and reuse across test files. Through hands-on trial and error and experimentation, I was able to grasp these concepts. Running tests using the command line allowed me to view detailed reports, which helped me identify minor errors in my logic.

This experience has highlighted the crucial role that testing plays in software development. In the disaster response system, every feature—from login authentication to real-time alerts—must function reliably under pressure. Pytest enables developers to verify this reliability efficiently and is therefore highly applicable in mission-critical applications.

References:
[5], [4], [3], [6]

2.5. Skills for Cybersecurity: Barnett Zheng

Through this project, I identified two critical skills from the SFIA framework relevant to cybersecurity:

Key Technical Skills

- **SCTY (Network Security)** [7]
  Securing communication channels in the disaster response system required:
  - Implementing firewall rules to restrict unauthorized access
  - Encrypting data transmissions using TLS to ensure confidentiality
  - Deploying intrusion detection systems (IDS) to monitor for suspicious activity
- **SCTY (Data Protection)** [8]
  Ensuring the privacy and integrity of user data involved:
  - Utilizing AES encryption for sensitive personal information storage
  - Implementing access controls to restrict unauthorized data retrieval
  - Performing regular security audits to detect vulnerabilities

Skill Development through Collaboration

The team environment enhanced these cybersecurity skills by:

- **Interdisciplinary Insights**: Working with software engineers helped me align security mechanisms with application logic, ensuring seamless integration.
- **Incident Response Drills**: Collaborating with the team in security simulations improved my ability to detect and mitigate threats quickly.
- **Knowledge Sharing**: Gained practical experience with GitHub security features, such as dependency scanning and secret detection.

Areas for Improvement

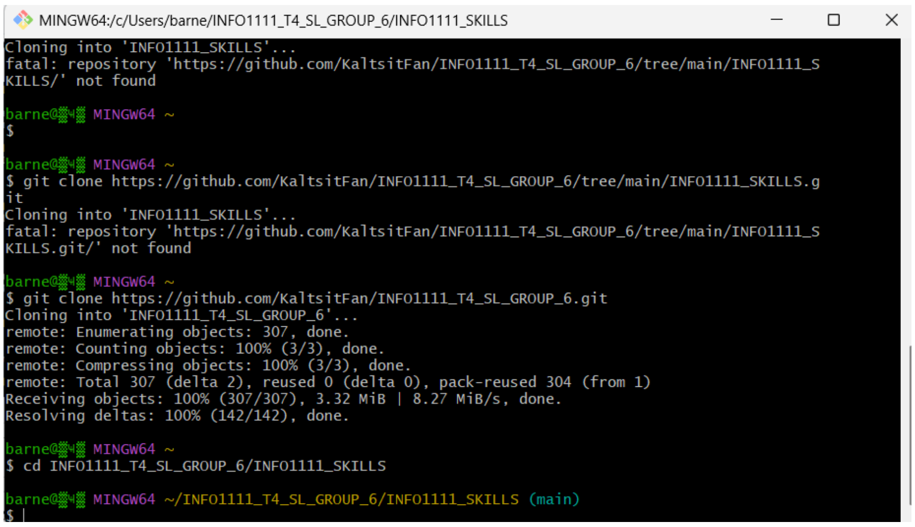
Through my work on the disaster response system, I've identified several cybersecurity skills that require further refinement:

- **USUP (Incident Response)** [9]: Need to enhance my ability to analyze and react to security breaches in real time, particularly in high-pressure situations.
- **RESL (Risk Assessment)** [10]: Improve my ability to evaluate and prioritize security threats, ensuring that mitigation efforts focus on the most critical risks.

2.3.1. Git Response

1. Clone the repository

Clone the remote repository to your local computer:



MINGW64:/c/Users/barne/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS

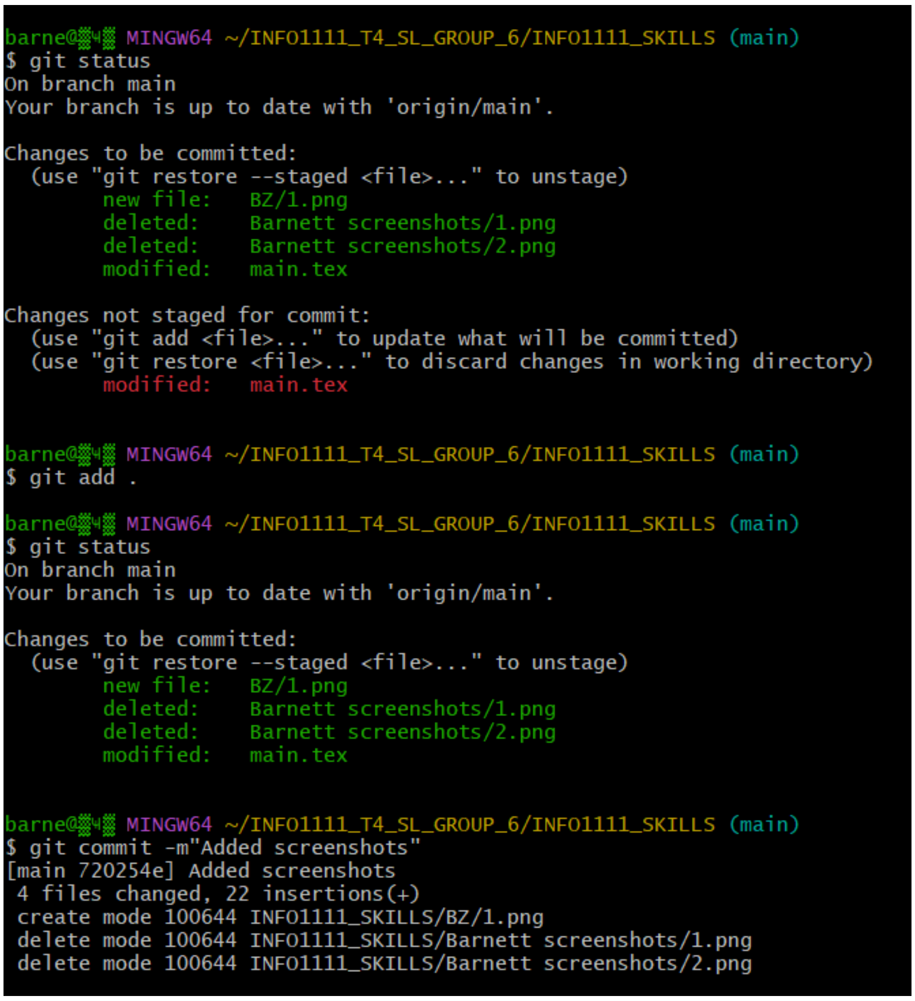
```
Cloning into 'INFO1111_SKILLS'...
fatal: repository 'https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6/tree/main/INFO1111_S
KILLS/' not found

barne@▒4▒ MINGW64 ~
$

barne@▒4▒ MINGW64 ~
$ git clone https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6/tree/main/INFO1111_SKILLS.g
it
Cloning into 'INFO1111_SKILLS'...
fatal: repository 'https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6/tree/main/INFO1111_S
KILLS.git/' not found

barne@▒4▒ MINGW64 ~
$ git clone https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6.git
Cloning into 'INFO1111_T4_SL_GROUP_6'...
remote: Enumerating objects: 307, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 307 (delta 2), reused 0 (delta 0), pack-reused 304 (from 1)
Receiving objects: 100% (307/307), 3.32 MiB | 8.27 MiB/s, done.
Resolving deltas: 100% (142/142), done.

barne@▒4▒ MINGW64 ~
$ cd INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS

barne@▒4▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ |
```

Figure 14: Cloning the remote repository

2. Standard Development Cycle

The regular workflow for making changes consists of these steps:

```
git add .                           # Stage all modified files
git commit -m "Descriptive message" # Commit changes locally
git status                          # Verify repository state
```



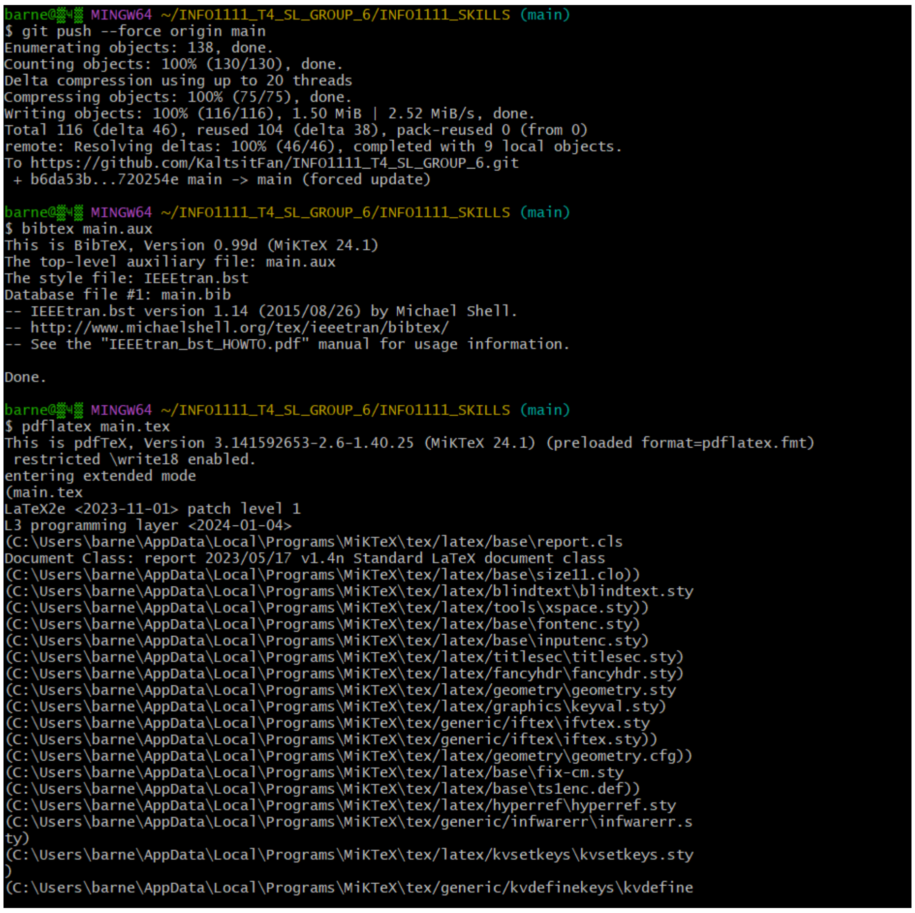
```
barne@▒▒▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   BZ/1.png
        deleted:    Barnett screenshots/1.png
        deleted:    Barnett screenshots/2.png
        modified:   main.tex

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.tex


barne@▒▒▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ git add .

barne@▒▒▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   BZ/1.png
        deleted:    Barnett screenshots/1.png
        deleted:    Barnett screenshots/2.png
        modified:   main.tex


barne@▒▒▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ git commit -m"Added screenshots"
[main 720254e] Added screenshots
 4 files changed, 22 insertions(+)
 create mode 100644 INFO1111_SKILLS/BZ/1.png
 delete mode 100644 INFO1111_SKILLS/Barnett screenshots/1.png
 delete mode 100644 INFO1111_SKILLS/Barnett screenshots/2.png
```

Figure 15: Git workflow: staging, committing

3. Document Compilation

Generate the project PDF through terminal commands:

```
pushing oringin main
bibtex main.aux       # Process citations
pdflatex main.tex     # Final compilation
```



```
barne@▒ч▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ git push --force origin main
Enumerating objects: 138, done.
Counting objects: 100% (130/130), done.
Delta compression using up to 20 threads
Compressing objects: 100% (75/75), done.
Writing objects: 100% (116/116), 1.50 MiB | 2.52 MiB/s, done.
Total 116 (delta 46), reused 104 (delta 38), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (46/46), completed with 9 local objects.
To https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6.git
 + b6da53b...720254e main -> main (forced update)

barne@▒ч▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ bibtex main.aux
This is BibTeX, Version 0.99d (MiKTeX 24.1)
The top-level auxiliary file: main.aux
The style file: IEEEtran.bst
Database file #1: main.bib
-- IEEEtran.bst version 1.14 (2015/08/26) by Michael Shell.
-- http://www.michaelshell.org/tex/ieeetran/bibtex/
-- See the "IEEEtran_bst_HOWTO.pdf" manual for usage information.

Done.

barne@▒ч▒ MINGW64 ~/INFO1111_T4_SL_GROUP_6/INFO1111_SKILLS (main)
$ pdflatex main.tex
This is pdfTeX, Version 3.141592653-2.6-1.40.25 (MiKTeX 24.1) (preloaded format=pdflatex.fmt)
 restricted \write18 enabled.
entering extended mode
(main.tex
LaTeX2e <2023-11-01> patch level 1
L3 programming layer <2024-01-04>
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/base\report.cls
Document Class: report 2023/05/17 v1.4n Standard LaTeX document class
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/base\size11.clo))
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/blindtext\blindtext.sty
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/tools\xspace.sty))
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/base\fontenc.sty)
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/base\inputenc.sty)
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/titlesec\titlesec.sty)
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/fancyhdr\fancyhdr.sty)
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/geometry\geometry.sty
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/graphics\keyval.sty)
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/generic/iftex\ifvtex.sty
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/generic/iftex\iftex.sty))
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/geometry\geometry.cfg))
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/base\fix-cm.sty
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/base\ts1enc.def))
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/hyperref\hyperref.sty
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/generic/infwarerr\infwarerr.s
ty)
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/latex/kvsetkeys\kvsetkeys.sty
)
(C:\Users\barne\AppData\Local\Programs\MiKTeX\tex/generic/kvdefinekeys\kvdefine
```

Figure 16: Compiling LaTeX document to PDF and pushing

Skills for Data Science: Link Lin

Through my work exploring data science applications, particularly in disaster response and predictive analytics, I identified two critical skills relevant to data science based on their practical importance:

Key Technical Skills

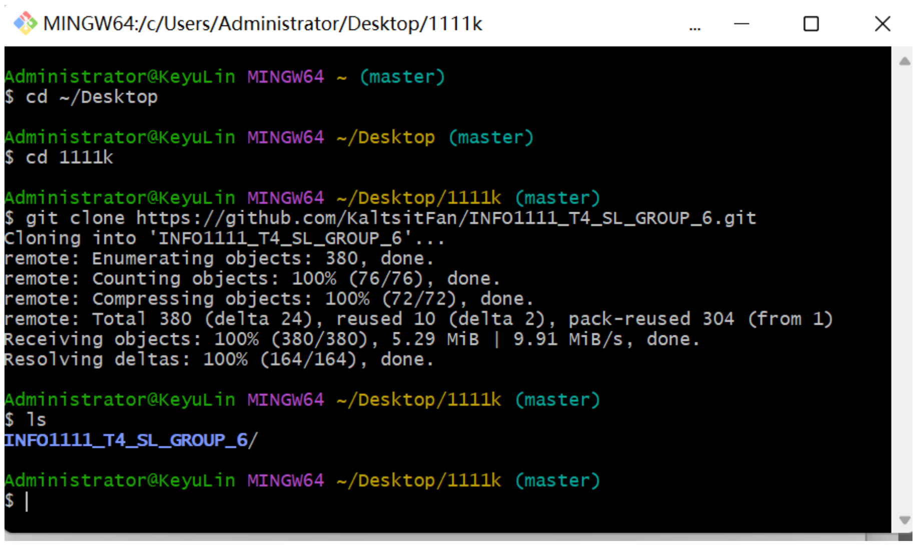
- **Data Integration**
  Effective data integration for disaster scenarios required:
  - Combining diverse datasets (e.g., satellite imagery, weather forecasts) using Python's `Pandas` library
  - Normalizing inconsistent formats for real-time visualization during emergencies
- **Predictive Modeling**
  Building robust predictive models for fire prevention involved:
  - Analyzing historical weather and topographic data with:
    - Machine learning algorithms (e.g., Random Forests)
    - Cross-validation to prevent overfitting
  - Simulating fire spread scenarios featuring:

∗ Temperature and humidity forecasting
∗ Wind speed impact modeling
∗ Risk area prioritization

2.3.2. git response

1. Clone the repository

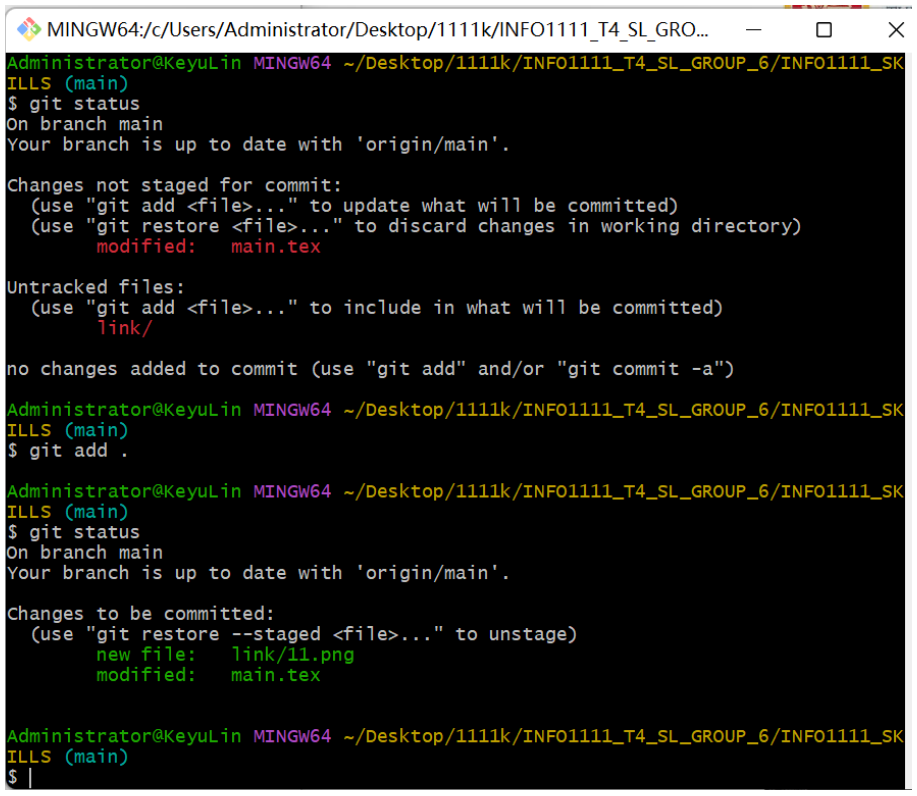
Clone the remote repository to your local computer:



```
Administrator@KeyuLin MINGW64 ~ (master)
$ cd ~/Desktop

Administrator@KeyuLin MINGW64 ~/Desktop (master)
$ cd 1111k

Administrator@KeyuLin MINGW64 ~/Desktop/1111k (master)
$ git clone https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6.git
Cloning into 'INFO1111_T4_SL_GROUP_6'...
remote: Enumerating objects: 380, done.
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 380 (delta 24), reused 10 (delta 2), pack-reused 304 (from 1)
Receiving objects: 100% (380/380), 5.29 MiB | 9.91 MiB/s, done.
Resolving deltas: 100% (164/164), done.

Administrator@KeyuLin MINGW64 ~/Desktop/1111k (master)
$ ls
INFO1111_T4_SL_GROUP_6/

Administrator@KeyuLin MINGW64 ~/Desktop/1111k (master)
$ |
```

Figure 17: Cloning

2. Standard Development Cycle



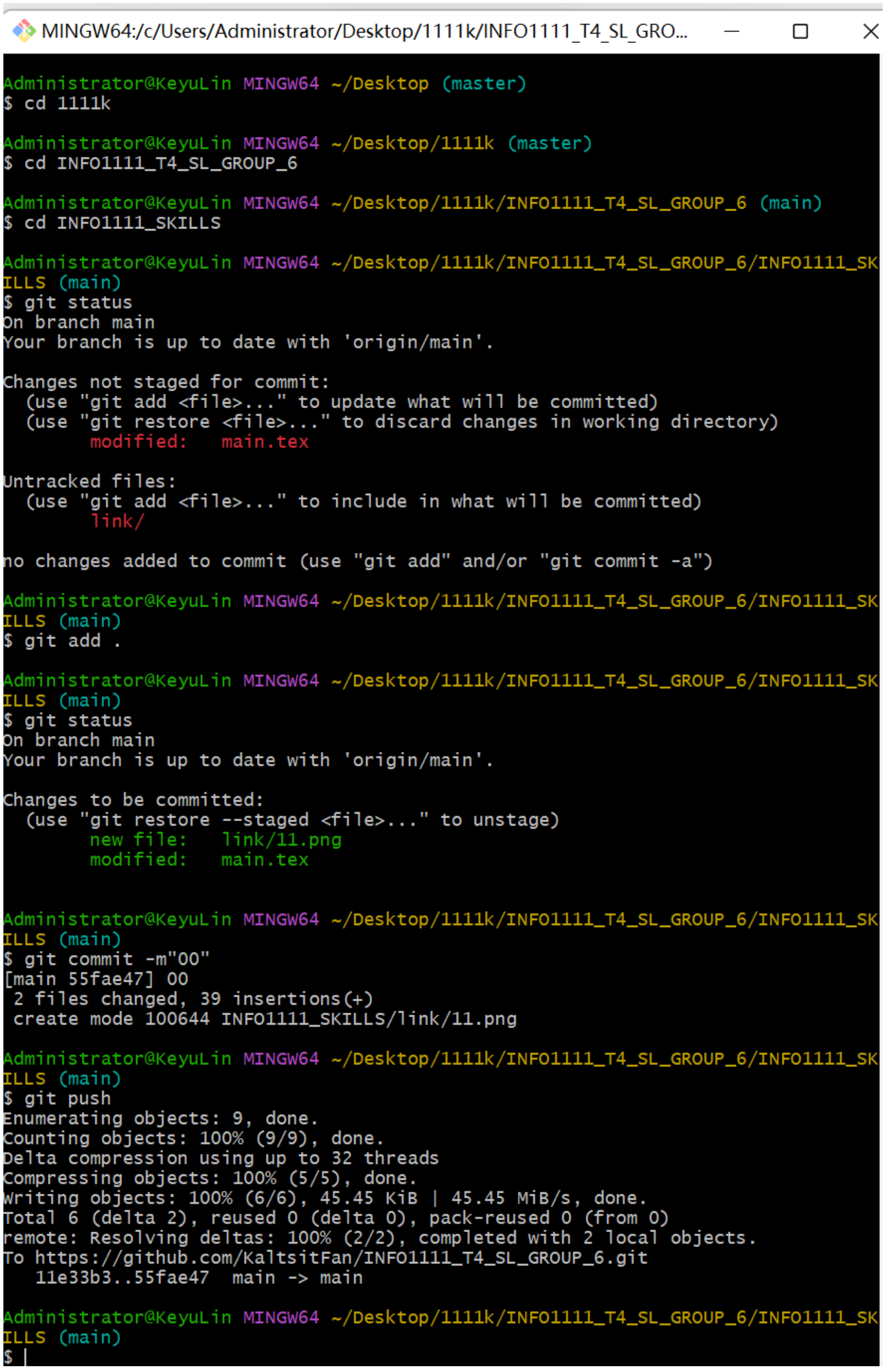
MINGW64:/c/Users/Administrator/Desktop/1111k/INFO1111_T4_SL_GRO...

```
Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        link/

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git add .

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   link/11.png
        modified:   main.tex


Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ |
```

Figure 18: add

3. Testing and Deployment

Execute tests and deploy through terminal commands:



MINGW64:/c/Users/Administrator/Desktop/1111k/INFO1111_T4_SL_GRO...

```
Administrator@KeyuLin MINGW64 ~/Desktop (master)
$ cd 1111k

Administrator@KeyuLin MINGW64 ~/Desktop/1111k (master)
$ cd INFO1111_T4_SL_GROUP_6

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6 (main)
$ cd INFO1111_SKILLS

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        link/

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git add .

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   link/11.png
        modified:   main.tex


Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git commit -m"00"
[main 55fae47] 00
 2 files changed, 39 insertions(+)
 create mode 100644 INFO1111_SKILLS/link/11.png

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 32 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 45.45 KiB | 45.45 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6.git
   11e33b3..55fae47  main -> main

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ |
```

Figure 19: push

Skill Development through Collaboration

The team environment enhanced these skills by:

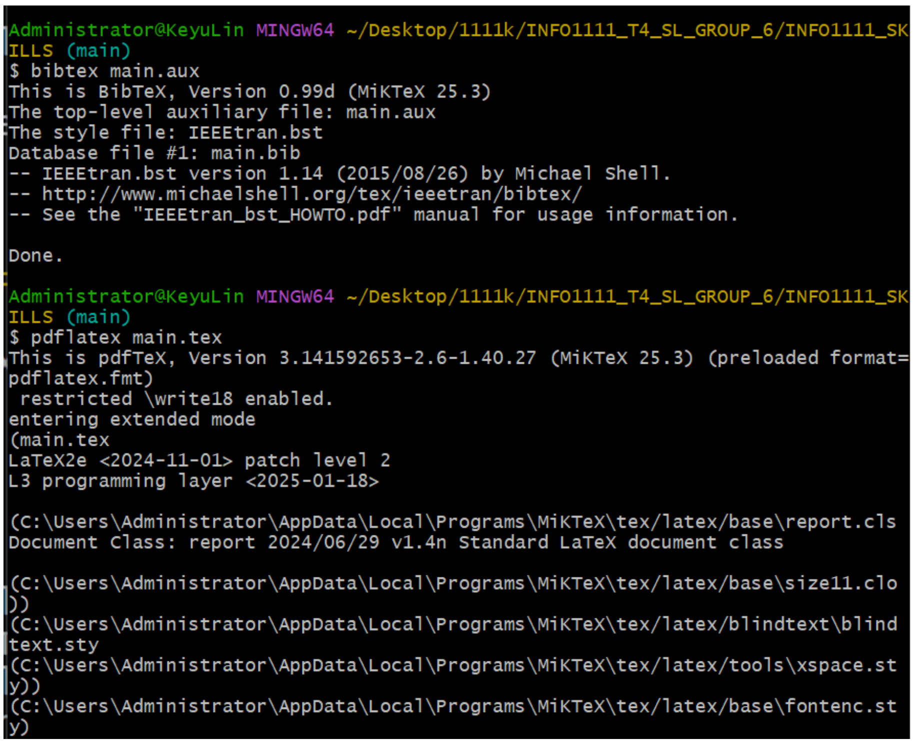
- **Cross-disciplinary input**: Feedback from software development teammates improved data pipeline efficiency, optimizing how integrated data fed into predictive models.

- **Group troubleshooting**: Collaborative debugging sessions refined model accuracy by addressing data preprocessing errors.
- **Tool adoption**: Learned Jupyter Notebook workflows from a teammate, enhancing my ability to prototype and visualize data integration outputs.

Areas for Improvement

Through my data science efforts, I've identified key areas for growth:

- **Data Quality Handling**: Need better proficiency in manual data cleaning techniques (e.g., handling missing values), as shown when inconsistent weather data skewed early predictions.
- **Model Interpretability**: Struggle to explain complex model outputs clearly; improving visualization skills with tools like Matplotlib or Seaborn will aid communication with non-technical stakeholders.



```
Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ bibtex main.aux
This is BibTeX, Version 0.99d (MiKTeX 25.3)
The top-level auxiliary file: main.aux
The style file: IEEEtran.bst
Database file #1: main.bib
-- IEEEtran.bst version 1.14 (2015/08/26) by Michael Shell.
-- http://www.michaelshell.org/tex/ieeetran/bibtex/
-- See the "IEEEtran_bst_HOWTO.pdf" manual for usage information.

Done.

Administrator@KeyuLin MINGW64 ~/Desktop/1111k/INFO1111_T4_SL_GROUP_6/INFO1111_SK
ILLS (main)
$ pdflatex main.tex
This is pdfTeX, Version 3.141592653-2.6-1.40.27 (MiKTeX 25.3) (preloaded format=
pdflatex.fmt)
 restricted \write18 enabled.
entering extended mode
(main.tex
LaTeX2e <2024-11-01> patch level 2
L3 programming layer <2025-01-18>

(C:\Users\Administrator\AppData\Local\Programs\MiKTeX\tex/latex/base\report.cls
Document Class: report 2024/06/29 v1.4n Standard LaTeX document class

(C:\Users\Administrator\AppData\Local\Programs\MiKTeX\tex/latex/base\size11.clo
))
(C:\Users\Administrator\AppData\Local\Programs\MiKTeX\tex/latex/blindtext\blind
text.sty
(C:\Users\Administrator\AppData\Local\Programs\MiKTeX\tex/latex/tools\xspace.st
y))
(C:\Users\Administrator\AppData\Local\Programs\MiKTeX\tex/latex/base\fontenc.st
y)
```

Figure 20: pdflatex

3. Submission contribution overview

For each submission, outline the approach taken to your teamwork, how you combined the various contributions, and whether there were any significant variations in the levels of involvement. (Target = ∼100-300 words).

3.1. Submission 1 contribution overview

As above, for submission 1 Kaffa DEmo2

3.2. Submission 2 contribution overview

As above, for submission 2

3.3. Submission 3 contribution overview

As above, for submission 3

Bibliography

[1] SFIA, “The global skills and competency framework for the digital world,” 2022, see https://sfia-online.org/en/sfia-8/all-skills-a-z.

[2] Websockets Library Documentation, “Websockets library documentation, version 10.3,” 2025, available: https://websockets.readthedocs.io/ (accessed May 21, 2025).

[3] SFIA Foundation, “Sfia 8 reference guide,” 2021, available: https://sfia-online.org/en/sfia-8 (accessed May 21, 2025).

[4] pytest dev, “Pytest documentation,” 2024, https://docs.pytest.org.

[5] B. Okken, *Python Testing with Pytest.* Pragmatic Bookshelf, 2017.

[6] The University of Sydney, “Referencing and citation styles: IEEE,” 2022, see https://libguides.library.usyd.edu.au/c.php?g=508212.

[7] National Institute of Standards and Technology (NIST), “Computer system security and privacy advisory meeting,” 2002, see https://csrc.nist.gov/CSRC/media/Events/CSSPAB-JUNE-2002-MEETING/documents/Lainhart-06-2002.pdf.

[8] European Union Agency for Cybersecurity (ENISA), “A trusted and cyber secure europe,” 2024, see https://www.enisa.europa.eu/sites/default/files/2025-02/A%20Trusted%20and%20Cyber%20Secure%20Europe%20-%20ENISA%20Strategy%20-%20Indicators.pdf.

[9] SANS Institute, “Incident response,” 2021, see https://www.sans.org/security-resources/glossary-of-terms/incident-response/.

[10] International Organization for Standardization (ISO), “Information security, cybersecurity and privacy protection,” 2022, see https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27005:ed-4:v1:en.