

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 | A | 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 | 2 |
| 2. | Individual Response | 3 |
| | 2.1. Skills for Computer Science: Kaffa Fan | 3 |
| | 2.2. Skills for SW Development: Jared Song | 6 |
| | 2.3. Skills for Cybersecurity: Barnett Zheng | 7 |
| 3. | Submission contribution overview | 9 |
| | 3.1. Submission 1 contribution overview | 9 |
| | 3.2. Submission 2 contribution overview | 9 |
| | 3.3. Submission 3 contribution overview | 9 |

1. Group Response

Our team consisted of four members from different computing majors: Computer Science, Data Science, Software Development, and Cyber Security. The primary objective of this project was to practice collaborative skills using GitHub rather than developing a complete solution. We effectively utilized GitHub's issue tracking system to clearly divide and allocate tasks according to our individual specialties. This structured approach facilitated cross-disciplinary support; for instance, the Software Development member helped the Cyber Security member resolve coding challenges, while the Data Science member guided the team through data-related aspects.

A significant challenge arose during documentation writing with LaTeX. Initially, we wrote LaTeX files directly without any real-time PDF previews. Due to this limitation, we frequently introduced syntax errors, such as incorrectly matched or entirely omitted commands like `\begin{itemize}` and `\end{itemize}`. These mistakes went unnoticed until compilation attempts, resulting in numerous difficulties and considerable debugging efforts later on. To mitigate this, we transitioned to using Visual Studio Code with LaTeX extensions, providing an Overleaf-like real-time PDF preview. This allowed us to immediately detect and correct errors as they occurred, significantly streamlining the documentation process and enhancing overall productivity.

Ultimately, by collaboratively addressing challenges and leveraging tools like GitHub and VS Code, we greatly improved our team's efficiency, communication, and task management capabilities.

2. Individual Response

2.1. Skills for Computer Science: Kaffa Fan

Personal Skill Reflection (Computer Science)

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:

1. Software Development (PROG)

According to the SFIA framework [1], software development is foundational to implementing various technological solutions within the computer science domain. Mastery of LaTeX as a report-writing tool requires proficiency in real-time debugging and structured document organization. **Skill Enhancement (Team Collaboration Details):**

Initially, I decided that the team would use a plain text editor to write LaTeX code because most of my teammates had not installed any latex environments. And Unlike Overleaf, these editors lacked real-time output previews, leading to frequent errors, such as missing commands like `\end{itemize}`, resulting in significant debugging time.

Later, by transitioning to Visual Studio Code (VS Code) and installing plugins like LaTeX Workshop and LaTeX Language Support, we enabled real-time compilation and error notifications. This substantially improved coding efficiency and code quality.

Areas for Further Improvement:

Currently, there is an over-reliance on real-time feedback from development tools. Moving forward, I aim to strengthen my proactive code review and logical self-checking skills to enhance code robustness.

2. Systems Integration and Build (SINT)

According to the SFIA framework [1], systems integration and build skills emphasize efficient integration of various modules to ensure smooth operation of the final system.

Skill Enhancement (Team Collaboration Details):

Initially, communication barriers existed within our team due to different personal communication habits. This caused vague task allocation and reduced efficiency.

To address these communication issues, the team adopted GitHub Issues for clear task division and tracking. Tasks such as "Frontend Interface Design" and "Data Interface Development" were specifically defined, allowing team members to clearly understand and take responsibility for their respective areas.

Areas for Further Improvement:

Technical discussions still contained excessive professional jargon, and unfamiliarity with various software and environments led to difficulties in comprehension among other team members. In the future, I should enhance my ability to explain complex technical concepts in simpler terms to improve cross-disciplinary communication effectiveness.

3. Git Response

1. Clone the repository

Clone the remote repository to your local computer:

```
git clone https://github.com/KaltsitFan/INF01111_T4_SL_GROUP_6
```

This picture illustrates the cloning process:

```

Administrator@9800X3D MINGW64 ~/Desktop/1111/INF01111_GROUP/INF01111_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/INF01111_GROUP/INF01111_SKILLS (main)
$ git clone ^[[200~https://github.com/kaltsitFan/INF01111_T4_SL_GROUP_6~
Cloning into 'INF01111_T4_SL_GROUP_6'...
fatal: protocol '^[[200~https' is not supported

Administrator@9800X3D MINGW64 ~/Desktop/1111/INF01111_GROUP/INF01111_SKILLS (main)
$ git clone https://github.com/kaltsitFan/INF01111_T4_SL_GROUP_6
Cloning into 'INF01111_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/INF01111_GROUP/INF01111_SKILLS (main)
$ ls
INF01111_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/INF01111_GROUP/INF01111_SKILLS (main)
$

```

Figure 1: 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/INF01111_GROUP/INF01111_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/INF01111_GROUP/INF01111_SKILLS (main)
$ git add .
warning: in the working copy of 'INF01111_SKILLS/main.aux', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INF01111_SKILLS/main.fls', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INF01111_SKILLS/main.log', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INF01111_SKILLS/main.toc', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'INF01111_SKILLS/main.blg', LF will be replaced by CRLF the next time Git touches it

Administrator@9800X3D MINGW64 ~/Desktop/1111/INF01111_GROUP/INF01111_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 INF01111_SKILLS/kaffa/clone.png
create mode 100644 INF01111_SKILLS/main.blg

Administrator@9800X3D MINGW64 ~/Desktop/1111/INF01111_GROUP/INF01111_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/INF01111_T4_SL_GROUP_6.git
To https://github.com/kaltsitFan/INF01111_GROUP.git
 c626c87..19c5bfc  main -> main

Administrator@9800X3D MINGW64 ~/Desktop/1111/INF01111_GROUP/INF01111_SKILLS (main)
$

```

Figure 2: 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.bb1  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 3: 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/iftex.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/tlenc.def))
```

2.2. 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 scenarios:

- 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.

2.3. 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)**

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)**

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

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

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.

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 = \sim 100-300 words).

3.1. Submission 1 contribution overview

As above, for submission 1

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>.