

# Team Charter – team 10x

---

## 1. Team Purpose and Objectives

### Purpose:

Our purpose is to better ourselves as software engineers and team members by creating high quality and maintainable software.

### Objectives (concrete/measurable):

- Ensure all members become T-shaped developers with experience in GitHub, HTML, CSS, and JS.
  - Develop skills to interact with other team members better.
- 

## 2. Team Values and Standards

### Core Values:

- Everyone has a fair voice
- Communicating with one another
- Respect each others work

### Guiding Principles:

- Daily document updates on the progress
    - Report what individual did, will do, and the blockers
  - Make sure most of the teammate participates
  - Take advantage on communications through Slack
  - Regular testing
- 

## 3. Roles and Responsibilities

Name	Primary Role	Secondary Role	Responsibilities
Nicholas Nurwinata	Project Lead	DevOps (CI/CD)	Schedule meetings, update documents, create pipelines and issues.

Ryan Kung	Project Lead	JavaScript developer	Coordinate meetings, update documents, write and review code
Fong Yu Lin	JS developer (interactions)	Designer	Design user interactions and project design
Audrey Fernandez	CSS developer (interface)	HTML dev	
Loreen Samaan	CSS developer (interface)	JavaScript developer	Make the page organized for users. Help out with JavaScript and maybe use it for styling as well.
Taha Qamar	JS developer	Code Reviewer	Write and review code
Alan Shapow	Communication Lead	Scheduler	Coordinate messages and reminders
Branden Sioson	QA/Test Lead	Contributor	Ensure functionality and report bugs
Alex Pan	DevOps (CI/CD)	Developer	Create build pipeline, automatic testing, generate documentation.
Eric Wang	HTML dev	Code Reviewer	Write HTML code, review code in general.
Johnson Chung	QA/Test Lead	Note Taker	Ensure functionality and report bugs

## 4. Strengths and Weaknesses

### Team Strengths:

- Lots of versatility, many people can fit many roles
- Willing to communicate
- Many people (11), with pairing structure for roles

### Weaknesses or Risks:

- Very limited meeting times where everyone can meet up
- Many different experience levels
- Not used to working in a team

### Improvement Plan:

- Solidifying meeting times
  - Work as a team more and collaborate more
    - Such as pairing within the team with another person
  - Try different approaches before resorting to AI
  - Get more advices from the TA and someone with experience
- 

## 5. Communication and Collaboration

**Tools:** GitHub, Slack, Zoom, Google Docs

**Meeting Schedule:**

- **Long Form Meetings:** Thursdays After class (mandatory besides Alan) (6:30 PM), Tuesdays (supplementary)
- **Short Form Meetings:** 3 times a week everybody replies to a thread in #meeting-minutes about what they've done, blockers, and what they plan to do.

**Response Time Expectation:** Within 24 hours

**Rules:**

- Notify if late or absent to long form meetings
  - Use GitHub Issues to track progress
  - Use proper branching rule sets
- 

## 6. Conflict Resolution Plan

- Step 1: Discuss privately
  - Step 2: Team meeting
  - Step 3: Escalate to TA if unresolved
  - Record decisions in a shared document
- 

## 7. Accountability and Enforcement

- Missed work without notice = 1 warning
  - Two warnings = reevaluate role or alert instructor
  - Everyone must contribute weekly
- 

## 8. Success Metrics

- Milestones completed on time
- Balanced workload
- Positive team communication
- All members feel respected and involved

## 9. Personal Goal

Fong Yu Lin - Become comfortable to work in a team and communicate with others about my working progress. Learn how to (1) design projects before (2) writing good code.

Taha Qamar - Experience software development in a team, focus on good/uniform coding practices rather than just having something run, experience reviewing and code testing process on GitHub for team projects

Branden Sioson - Engage and thrive in a team environment whilst further learning and improving upon coding skills.

Ryan Kung - Become more comfortable working in a team environment and be better at communicating.

Eric Wang - Improve communication skills to learn how to thrive in a software development environment while juggling other classes. Also, pick up the conceptual side of software engineering, like how to be a software architect.

Audrey Fernandez - Practice coding and software development in a team to prepare for real world settings.

Loreen Samaan - Work efficiently in a team environment using the skills we learned in lecture, and become more familiar with CSS and JavaScript.

Nicholas Nurwinata - improve leadership skills in software development, improve communication and work in a proper format. Being an Architect

Johnson Chung - Learn how to work in a team environment and how to use the tools needed for it. Also to be more fluent in Javascript, HTML, and CSS.

Alex Pan - Learn more about automatic build processes to understand open-source development better.

Alan Shapow - Get comfortable with communication in a team environment, and learn the fundamentals of software design to a point where I can successfully build, deploy, and maintain my own software projects moving forwards. Gaining familiarity with web development.

Signature: Alan Shapow