



Open Design & Technology

Empowering Designers to Embrace Technology

Introduction

Course Overview

Week 1

Why?

Modern problems require people who can:

- Understand **technology** deeply
- Think **systemically**
- Design **human-centered solutions**
- Work across **disciplinary boundaries**

Goals of the module

- Learning how to code and make circuits.
- Using design principles of technology to further your craft
- Understanding the approach of integrating technology into their design initiatives

Classroom Etiquettes & Attitudes

- Having fun is NOT optional!
- It can be a **difficult** module for some. Align your personal goals with how far you can stretch.
- Being **curious** and asking a lot of questions is mandatory.
- Take as many **notes/diagrams** as you can.
- Food & Beverages are not allowed in the classroom and its important to leave it in tidy and clean condition.

Assessment

- 2 Assessments
- Holi Break | Semester End
- Assessment 1 : Understanding of concepts & application
- Assessment 2 : Creative endeavor
- Module Blueprint

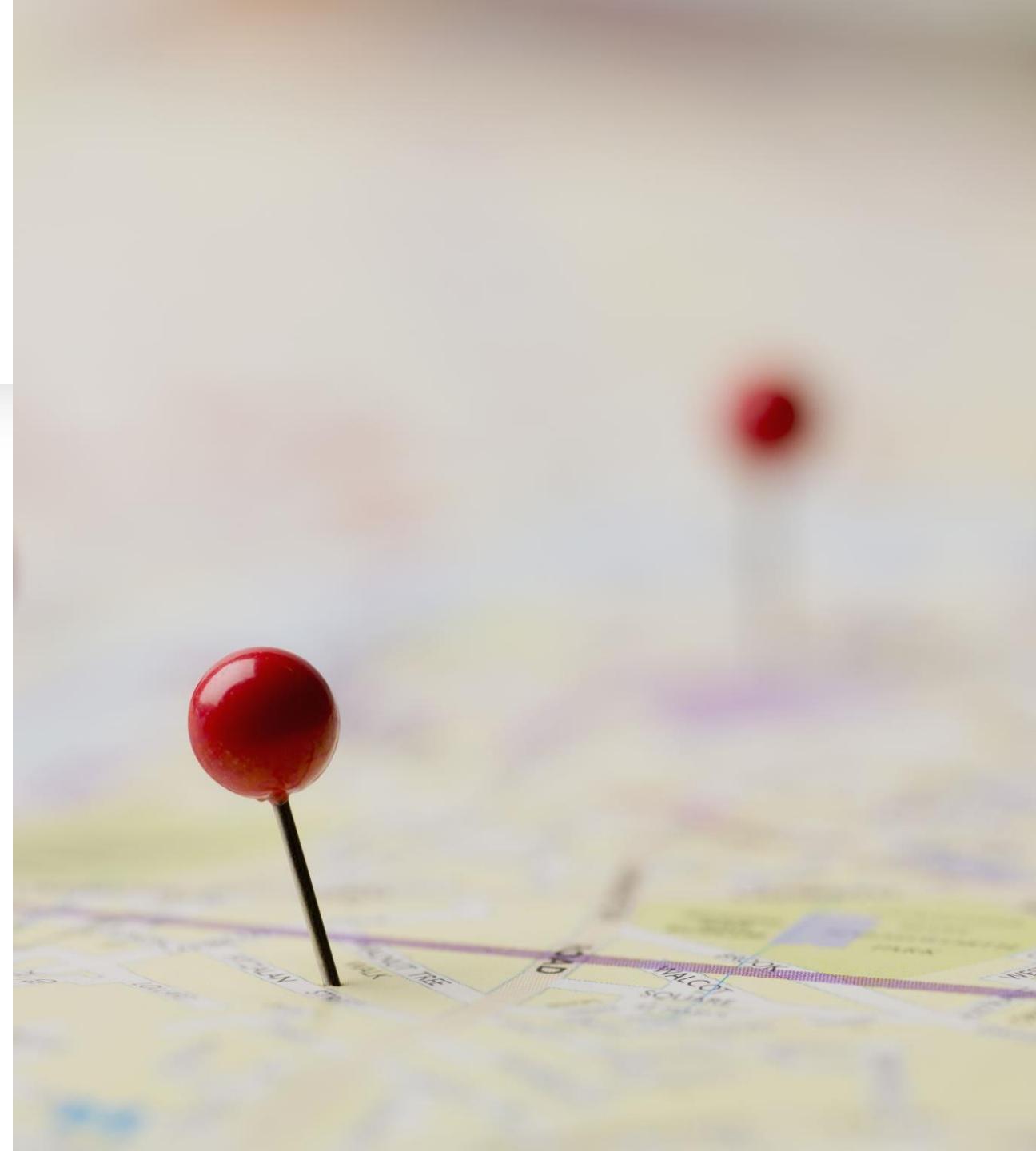
Individual evaluation

A black and white photograph of a young plant with two large, veined leaves growing out of a mound of dark, textured soil. The background is a solid, dark gray.

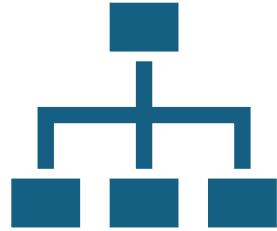
Group Formation

How do you name your files??

- Poster_final.psd
- Poster_final2.psd
- Poster_final_revised.psd
- Poster_final_revised_last.psd
- Poster_final_revised_last_FINAL.psd



Core Problem



Design Problems We All Face

- Overwriting files
- Losing old versions
- Confusion in group work
- “Who changed this?”
- No clean history of work



Git was invented to solve exactly this.

Git & GitHub

An Introduction

Git

- **Git is a version control system** that helps track changes in files over time.
- It allows multiple people to work on the same project, keep a history of edits, and go back to earlier versions if needed.

Git records:

- What changed
- Who changed it
- When it changed



GitHub

GitHub helps you:

- Work in teams
- Show process (not just final output)
- Build credibility
- Participate in open design projects
- Learn industry workflows early

Not just code, can store:

- Posters
- UI designs (Figma)
- Sketches
- Documents etc.

GitHub Terminologies



Repository (Repo)

A **Repository** is a central place where all the files of a project live, along with their history (changes over time).

It can contain:

- Files (text, images, code, etc.)
- Folders
- Previous versions of files

Imagine a Google Drive folder for a group project. All documents, images, and updates are stored there, and everyone can see the latest version.

Branch

A **Branch** is a separate line of work inside a repository where you can make changes **without affecting the main version**.

The default branch is usually called **main**.

Rough Draft : You photocopy the original document and experiment on the copy.
The original stays safe until you decide to replace or merge it.

Pull Request (PR)

A **Pull Request** is a request to merge changes from one branch (or fork) into another branch, usually into main.

It allows:

- Review
- Discussion
- Approval before merging

You finish your assignment and submit it for review.
The teacher checks it and decides whether to accept your changes.

Fork

A **Fork** is a personal copy of someone else's repository, created in your own GitHub account.

You can freely experiment without affecting the original repository.

You photocopy a book so you can highlight, write notes, or modify it without touching the original book in the library.

Term	Think of it as
Repository	Project folder
Branch	Rough draft
Fork	Personal copy
Pull Request	Submission for review



Activity 1 :

Repository

Creating a Repo | Adding
files | Editing files |
Branching



Activity 2 : Fork, Pull Request

"Issues??" | "Protecting the main" (Homework)