

WEEK 3: FRONTEND DEVELOPMENT WITH NEXT.JS

UML CLOUD COMPUTING CLUB

AGENDA

- Week 2 Recap
- Git Workflow
- What is Frontend Development?
- Introduction to HTML, CSS, and JavaScript.
- Introduction to React.js
- Introduction to Next.js

WEEK 2 RECAP

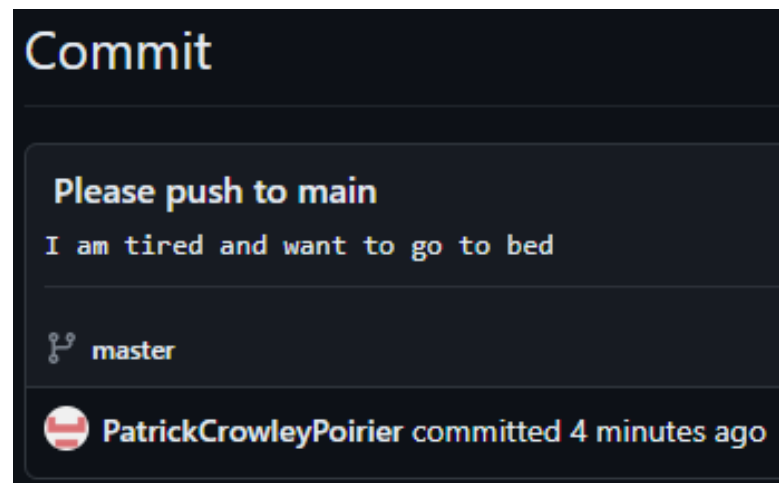
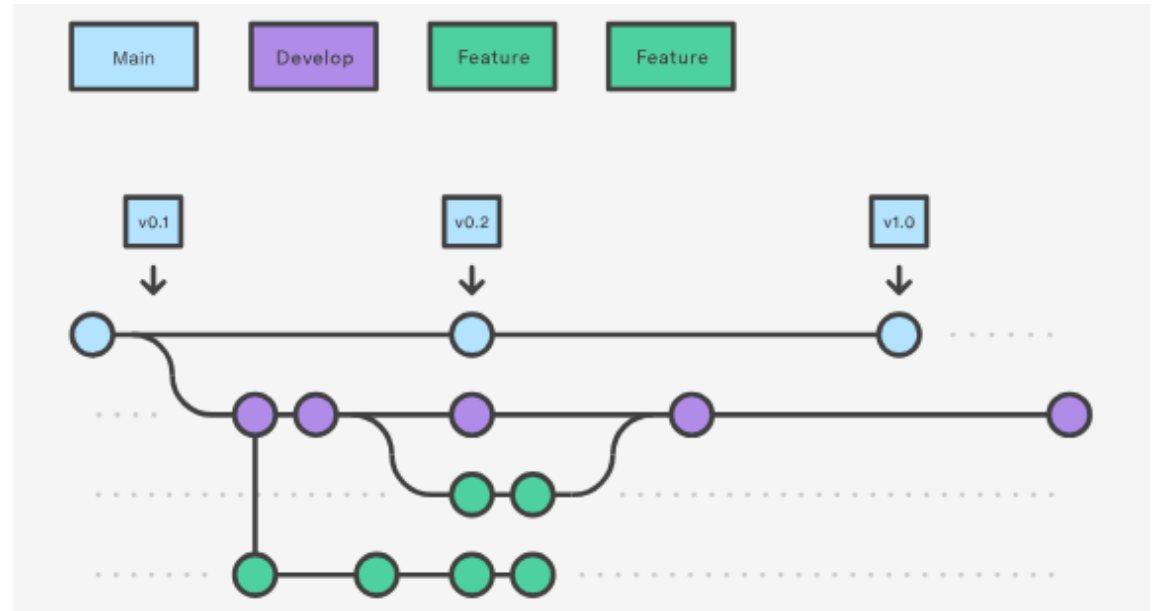
- Project Planning
 - Agile methodologies: Scrum and Kanban
 - Importance of effective planning
 - User stories and epics
- Software Engineering Practices
 - Code quality and readability
 - Git as a version control system
 - Testing: Unit, Integration, End-to-End
- Introduction to DevOps
 - Definition and significance
 - CI/CD pipelines
 - DevOps tools: Jenkins, Docker, Kubernetes
- Hands-On Activity:
 - Fork the UniPath.io project
 - Setup a CI/CD pipeline with feature branches

GIT: BASIC GIT COMMANDS

- **git clone:** Clone a repository
- **git add:** Stage changes
- **git commit:** Commit changes
- **git push:** Push changes to remote
- **git pull:** Pull changes from remote

GIT: GITFLOW WORKFLOW

- **What is Gitflow?**
 - A branching model for Git that defines a structured workflow.
- **Branches Involved**
 - **main**: The production-ready branch.
 - **develop**: The development branch, contains the latest delivered development changes.
 - **feature**: Branches for new features.
 - **release**: Branches for release tasks.
 - **hotfix**: Branches for quick fixes.
- **Workflow Steps**
 - Develop features in feature branches.
 - Merge feature branches into develop.
 - Merge develop branches into main
- **How We Use Gitflow in UniPath.io**
 - All feature development will be done in feature branches.
 - Code reviews before merging into develop.
 - Code reviews before merging into main



DEMO: GITFLOW WORKFLOW FOR UNIPATH.IO PROJECT

- Let's walk through a gitflow workflow together
- Follow “Gitflow Workflow for UniPath.io” tutorial



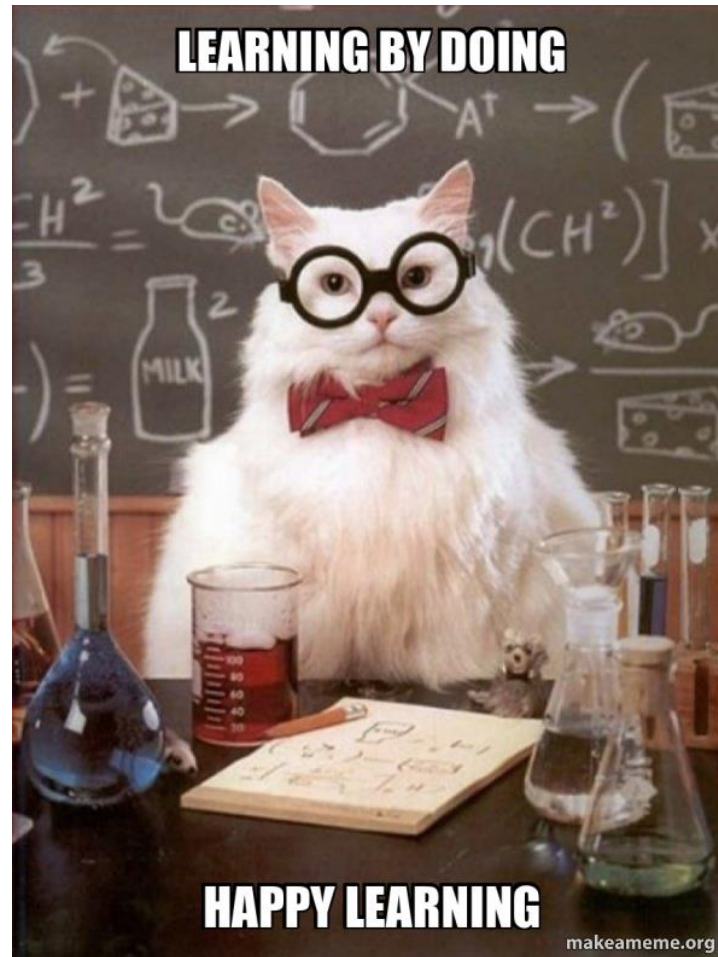
WHAT IS FRONTEND DEVELOPMENT?

- Definition
 - Frontend Development is the practice of converting data to a graphical interface for user interaction.
- Core Technologies
 - **HTML**: Structure of the web pages.
 - **CSS**: Styling and layout.
 - **JavaScript**: Interactivity and dynamic content.
- Responsibilities
 - Creating visually engaging user interfaces.
 - Ensuring cross-browser compatibility.
 - Implementing responsive design.
- Role in Web Development
 - Works in tandem with Backend Development to create a complete web application.
 - Frameworks and Libraries
 - React, Angular, Vue.js for more complex UI/UX.



DEMO: HTML, CSS, & JAVASCRIPT

- Let's walk through the basics of HTML, CSS, & JavaScript
- Follow “W3Schools: HTML, CSS, JS” tutorial



INTRODUCTION TO REACT.JS

- What is React.js?
 - A JavaScript library for building user interfaces.
- Who Developed It?
 - Developed and maintained by Facebook.
- Component-Based Architecture
 - Build encapsulated components that manage their own state.
- Virtual DOM
 - Optimizes rendering and improves performance.
- JSX Syntax
 - Write HTML-like syntax within JavaScript.
- Popular Use Cases
 - Single Page Applications (SPAs)
 - Mobile Applications (via React Native)

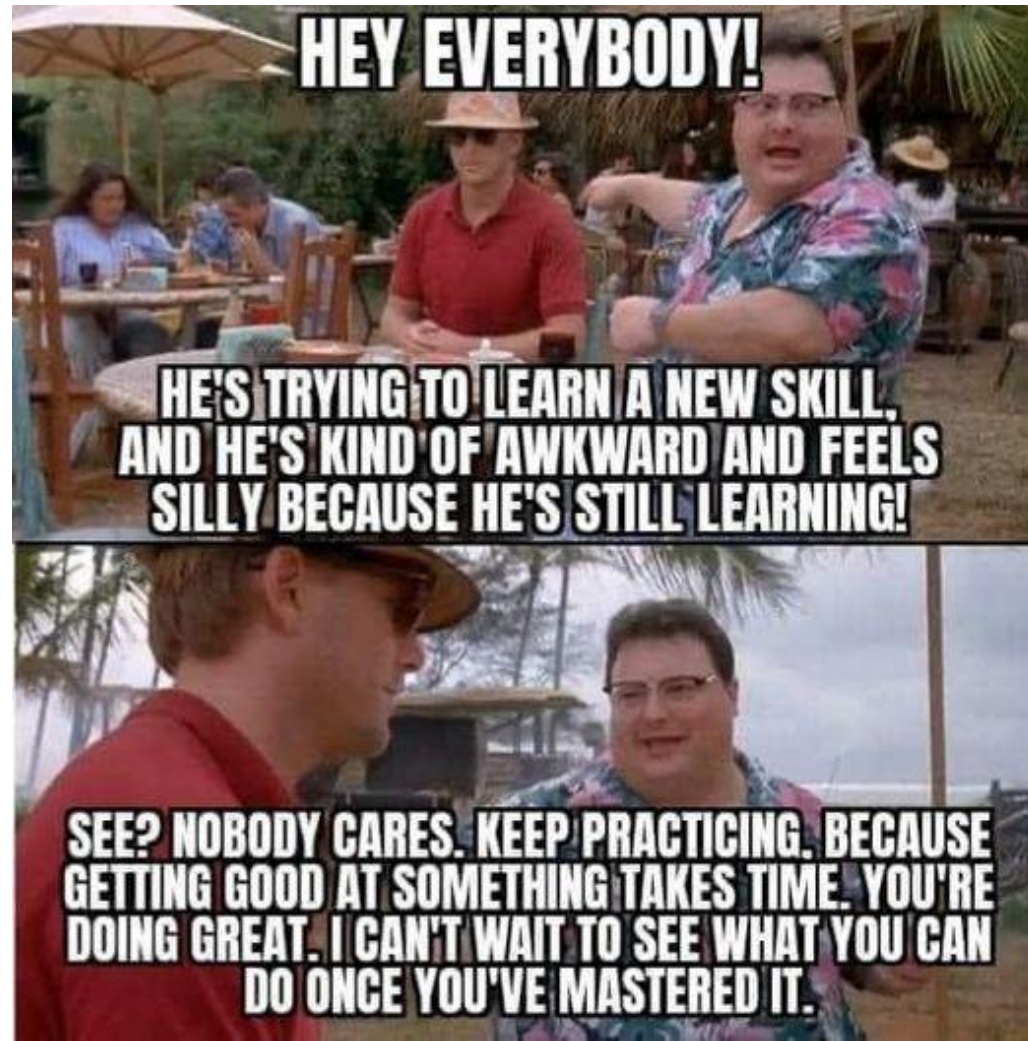


React JS Explained In 10 Minutes: <https://www.youtube.com/watch?v=s2skans2dP4>



DEMO: INTRO TO REACT

- Let's complete the official React.js quick start and tic-tac-toe tutorial



INTRODUCTION TO NEXT.JS

- What is Next.js?
 - A React framework for server-rendered JavaScript applications.
- Who Developed It?
 - Developed and maintained by Vercel.
- Server-Side Rendering (SSR)
 - Enables better SEO and initial page load performance.
- Static Site Generation (SSG)
 - Pre-renders pages at build time.
- API Routes
 - Built-in support for API endpoints.
- File-based Routing
 - Automatic routing based on the file structure.



Next.js – The Basics: https://www.youtube.com/watch?v=__mSgDEOyv8



DEMO: INTRO TO NEXT.JS

- Let's generate a new Next.js project and explore how it works

