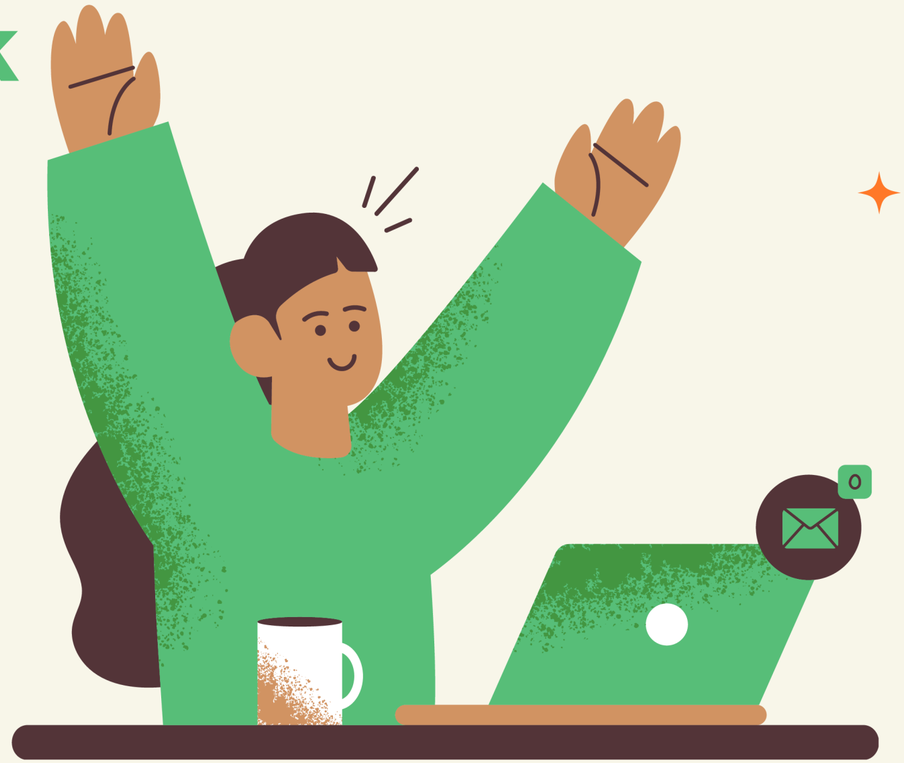
Cloud-based Full-Stack Web Development

KOCSEA & KSEA TG C-7 CIT workshop 1



April 19, 2025, 11 AM - 2 PM EDT

Clark Ngo
Nate Seon
Anh Bui
Harisson Le
Sam Chung

CIT-virtual-workshop

Speaker

Clark Jason Ngo is a software engineer and educator specializing in full-stack development, cloud computing, and scalable system design. He holds a Master of Science in Computer Science from the City University of Seattle and is currently pursuing a Master of Business Administration at the same institution.

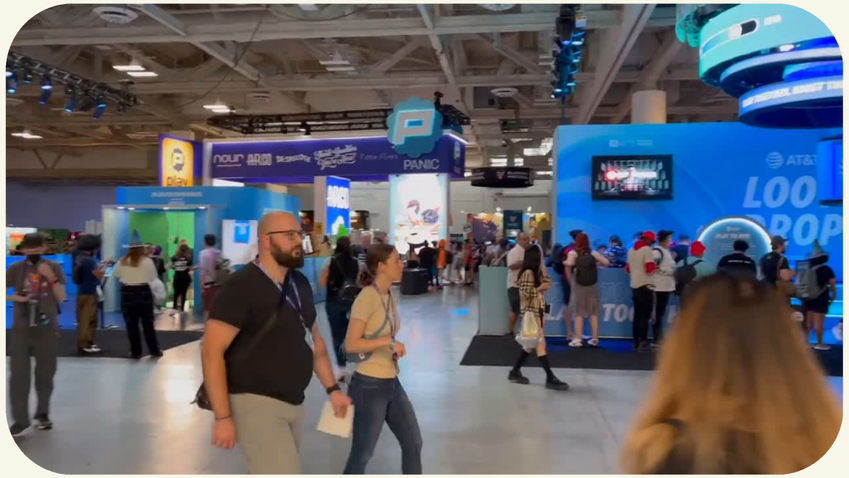
With extensive experience in software engineering, Mr. Ngo has worked at eBay, CloudEagle.AI, and Worldwide American, developing scalable and high-performance web applications using Java, Spring Boot, React, Python, and AWS. His expertise includes retrieval-augmented generation (RAG), machine learning workflows, and cloud infrastructure automation.

He has also been a Technical Program Manager, leading initiatives to train military veterans in full-stack development, cloud computing, and system administration. A dedicated educator and mentor, Mr. Ngo is passionate about bridging the gap between academia and industry, equipping aspiring developers with practical skills to build real-world applications.

Thomson Reuters™ S&P ebay

Fun


Your team has slain Baron Nashor!
GMC Zev
12
12
Bandle City Champion
13


PANIC

Workshop Assistants

Kina Bui

Nate Seon

Harisson Le

Sponsor

Sam Chung

Workshop Goal

By the end of the session, students will build a working To-Do application using:

React (Frontend)

Express.js (Backend)

MongoDB Atlas (Database)

Workshop Agenda (3 Hours)

0:00 - 0:20 Welcome and Setup

- Introductions
- Intro to full-stack development
- Tools needed

0:20 - 1:00 React Fundamentals

- What is React and why use it?
- Intro to JSX, components, and rendering
- Functional components vs class components
- Props and useState

1:00 - 1:45 Build Todo App (Frontend Only)

- File/project structure in Vite/React
- TodoForm and TodoList components
- Lifting state up to App.jsx
- Handling form input and rendering a list
- Managing local state (add/delete todos)

1:45 - 2:20 Express Backend + MongoDB

- Create basic Express server
- Install Mongoose & connect to MongoDB Atlas
- Define Todo schema/model
- Create basic routes: GET /api/todos, POST /api/todos
- Test with Postman or browser

2:20 - 2:45 Full-Stack Integration

- Axios: fetch data from backend
- Handle form submission with POST
- Use useEffect to load todos on page load
- CORS or Vite proxy setup

2:45 – 3:00 | Wrap-Up & Next Steps

- Demo final app
- Optional features (delete todo, mark complete)
- Deployment ideas
- Share resources for continued learning
- Q&A and feedback

What is Full-Stack Development?

🧱 **Full-Stack = Frontend + Backend + Database**

Frontend (Client-side)

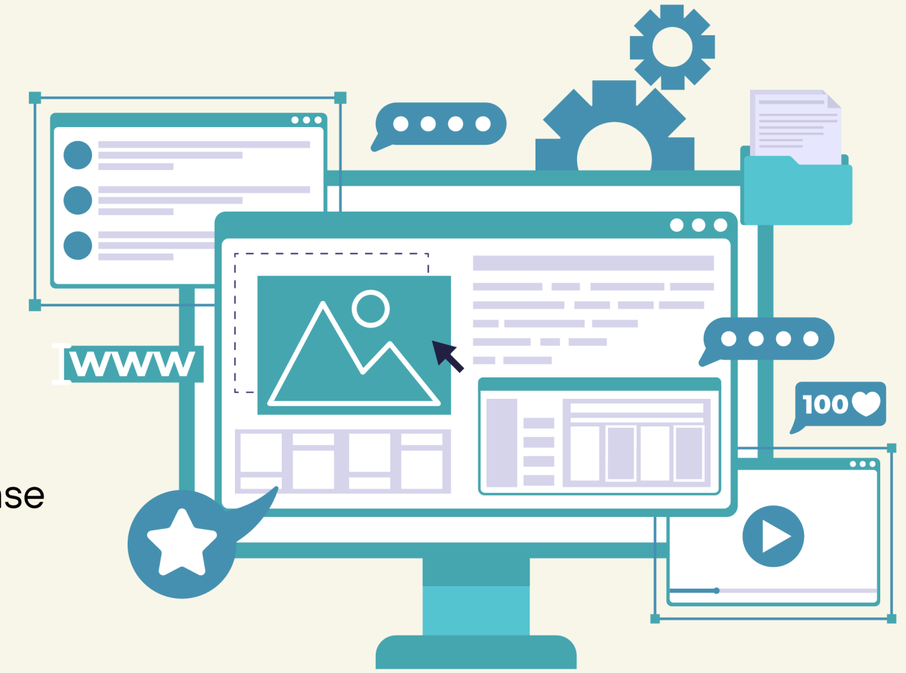
- What the user sees and interacts with
- Built with HTML/CSS/JS or frameworks like React

Backend (Server-side)

- Logic that handles requests, processes data, talks to the database
- Usually written in JavaScript, Java, Python, Ruby, etc.

Database

- Where data is stored
- Examples: MongoDB, MySQL, PostgreSQL



Tools Used

👷 Environment – GitHub Codespaces

- Cloud-based VS Code in the browser
- No local install required
- Pre-installed Node.js, Git, etc.
- Great for beginners and workshops

🖼️ Frontend – React + Vite

- React: For building UI with components
- Vite: Fast React development environment

🛠️ Backend – Express + Node.js

- Node.js: JavaScript runtime on the server
- Express: Lightweight framework to build APIs

🗃️ Database – MongoDB Atlas

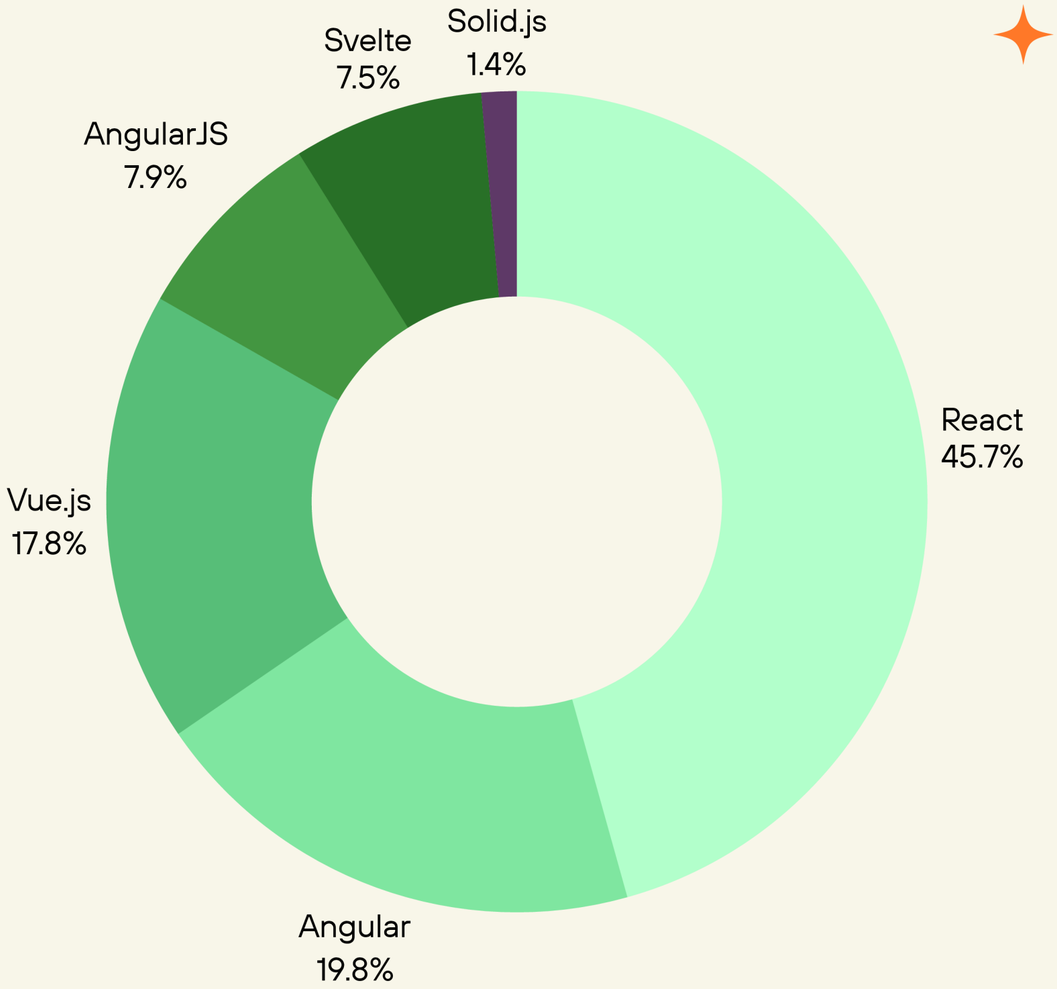
- Cloud-hosted NoSQL database
- Easy to connect from backend

🔗 Others

- Axios – For HTTP requests (frontend backend)
- Mongoose – MongoDB modeling
- Browser – To test API routes

React is Popular

- Highest job demand
- Component-Based - resusable components
- Dynamic and Interactive - updates automatically
- Easy Statement Management - useState, useEffect
- Huge Community and Ecosystem



2024 Stack Overflow Developer Survey

In May 2024, over 65,000 developers responded to our annual survey about coding, the technologies and tools they use and want to learn, AI, and developer experience at work. Check out the results and see what'...

stackoverflow

React Fundamentals

What is React?

- Why we use React (interactive UIs, component-based, reusable)
- Compare traditional HTML/JS vs. React-based apps
- Real-world analogy: LEGO blocks (components)

Hello World in React

- Intro to JSX (HTML-in-JavaScript)
- index.html and
- index.jsx and ReactDOM.createRoot()

React Components

- Functional components vs. class components (focus on functional)
- Props (pass data like function parameters)

React State & Events

- Intro to useState
- Button click

Tools Used

Feature	HTML/CSS/JS	React
Structure	Write everything in separate files: .html, .css, .js	Uses JSX: HTML + JS in one file (component)
Reusability	Rewriting similar code for each section	Wrap logic + UI into reusable components
Dynamic UI	Manually manipulate DOM with document.querySelector, etc.	UI auto-updates using state and hooks
Data Handling	You manually manage data, events, and UI syncing	React manages state and re-renders automatically
Integration	More boilerplate when connecting to APIs or managing updates	Easy API calls with useEffect, Axios, etc.

Questions?