

# Fullstack Development

# Self Evaluation

## Roadmap

<https://roadmap.sh/full-stack>

## Form

<https://forms.office.com/r/ZGzaZM9yxm>

# Preflight Project

# Objective

Develop "end-user-ready" full-Stack application in 1 months.

- Ideas

# Why?

- Give students experience of the overview of the full-stack technology.
  - Before going into "real" project.
- Want smaller groups ➡ better learning experience.
- Give students more chance to iterate ➡ better coding decision in the project .

# Requirement

- Full stack technology
  - Anything that you can run on local development environment.
  - No cloud service (e.g. Firebase)
  - If using JavaScript frameworks, use TypeScript.
- Automated Testing
  - API testing
  - UI testing

# Requirement (cont)

- Deployment
  - Deploy on CPS virtual machine (Ubuntu).
  - No Netlify, Vercel, Railway
- Source code repository
  - GitHub
  - GitLab
- CI/CD
  - Github action

# Sounds Impossible?

Acually, it is not that hard. Check out [last year's projects](#).

(Check `G2`, `G9`, `G11` )



# Due date

| Sunday 10 August 2015

## Deliverables

- URL to your deployed application
- URL to your source-code repositories
- URL to your VDO
  - File-downloadable (ผมขอเก็บไฟล์)
  - ความยาวไม่เกิน 5 นาที (Please)

# VDO Checklists (1)

- Introduction
  - แนะนำสมาชิกในกลุ่ม
  - ทำแอปอะไร สาธิตการทำงาน (Show me your CRUD!)
  - สรุป Technology Stack

## VDO Checklists (2)

- Database
  - ใช้ Database อะไร
  - มีโครงสร้างข้อมูลเป็นอย่างไร (e.g. แสดงหน้าใน Dbeaver)
  - ใช้ ORM อะไร
  - มีอะไรที่อยาก Show เทพ (แตกต่างจากที่สอน)

# VDO Checklists (3)

- Backend
  - ใช้ Framework อะไร
  - มี Endpoint อะไรบ้าง (สามารถอธิบายจาก Code หรือเขียน Diagram)
  - มีอะไรที่อยาก Show เทพ (แตกต่างจากที่สอน)
- Frontend
  - ใช้ Framework อะไร
  - อธิบาย Structure ของ Code คร่าวๆ
  - มีอะไรที่อยาก Show เทพ (แตกต่างจากที่สอน)

## VDO Checklists (4)

- Testing
  - ใช้ Framework อะไร
  - มี การ Test API อะไรบ้าง
  - มี การ Test UI อะไรบ้าง
  - มีอะไรที่อยาก Show เทพ (แตกต่างจากที่สอน)

# VDO Checklists (5)

- Deployment
  - ใช้ Technology อะไร (e.g. Docker)
  - อธิบาย Step ของการ Deployment
    - แสดง `Dockerfile` , `docker-compose.yml`
    - อธิบายการสร้าง Image/Container, etc.
  - มีอะไรที่อยาก Show เทพ (แตกต่างจากที่สอน)

## VDO Checklists (6)

- CI/CD
  - Show ว่าพอ Commit ใน Repository แล้วแอปอัพเดทได้












# ถ้าอยากใช้ Server ของตนเอง

การ Deploy ระบบของ Prelight Project ไม่จำเป็นต้องใช้ CPE Server เสมอไป หากมี Server ของตนเอง ก็สามารถนำมาใช้ได้เช่นกัน โดยมีเงื่อนไขดังต่อไปนี้

- ต้องเป็น Server หรือ Virtual Machine ที่เรามีสิทธิ์เป็นแอดมินเต็มรูปแบบ
- ไม่สามารถใช้บริการ Cloud ที่เป็นลักษณะ Fully Managed Deployment Automation เช่น Vercel, Railway, หรือ Cloudflare
- ต้องใช้เทคโนโลยี Container ตัวอย่างเช่น Docker หรือ Kubernetes
- ต้องมีการเชื่อมโยงกับ GitHub Actions เพื่อให้รองรับระบบอัตโนมัติสำหรับการ Build และ Deploy
  - สามารถใช้ Self-hosted Runner หรือจะใช้ Runner ของ GitHub ก็ได้
- ต้องมี Public URL สำหรับเข้าถึง Application

# Grade

- 10%
- Score distribution (see image)

Quiz - frontend development	2. สามารถพัฒนาระบบแอปพลิเคชันด้วยเฟรมเวิร์กด้านของฟร... x	x	10	
Quiz - RESTful API development	3. สามารถพัฒนาระบบบริการ RESTful API ด้วยเฟรมเวิร์กด้านของฟ... x	x	10	
Database design lab.	4. สามารถออกแบบโครงสร้างฐานข้อมูลได้อย่างเหมาะสม (AbL... x	x	10	
Testing lab.	5. สามารถทดสอบการทำงานของระบบแอปพลิเคชันด้วยเครื่... x	x	10	
การนำเสนอปลายภาค (final presentation)	1. อธิบายหน้าที่ และหลักการทำงานขององค์ประกอบต่าง ๆ ของ... x	x	10	
การสอบวัดความรู้ภาคทฤษฎี	1. อธิบายหน้าที่ และหลักการทำงานขององค์ประกอบต่าง ๆ ของ... x	x	10	
Full stack application project - frontend	2. สามารถพัฒนาระบบแอปพลิเคชันด้วยเฟรมเวิร์กด้านของฟร... x	x	20	
Full stack application project - RESTful API	3. สามารถพัฒนาระบบบริการ RESTful API ด้วยเฟรมเวิร์กด้านของฟ... x	x	10	
Full stack application project - deployment	6. สามารถติดตั้งแอปพลิเคชันที่พัฒนาโดยใช้เทคโนโลยีที่เหมาะสม... x	x	10	

[+ Add Assessment Tool](#) ผลรวมร้อยละการประเมิน (Total Evaluate) 100 %

## Extra credit

- Nice-looking UI design
- More complex data relationship (e.g. join tables)
- Includes Authentication / Authorization
- Use different stacks than the class example.

# Timeline (1)

Date	Day	Topics
24 Jun	Tue	Modern Javascript / Typescript
27 Jun	Fri	TypeScript
01 Jul	Tue	Database
04 Jul	Fri	Database
08 Jul	Tue	Backend

## Timeline (2)

Date	Day	Topics
11 Jul	Fri	<i>no class</i>
15 Jul	Tue	Backend
18 Jul	Fri	Frontend
22 Jul	Tue	Testing
25 Jul	Fri	Deploy
10 Aug	Sun	Deadline

# Group assignment

| Go to Mango Canvas

# Tools

- VSCode
  - Prettier
  - ESLint, ES7
  - Quokka.js
- Docker Desktop
- Node.js
- Git

# Tools

- Database client
  - Dbeaver
- API Testing tool
  - Insomnia
  - Postman
  - Thunder client (VSCode extension)



# AI Assistant

Disclaimer: Nirand's take

- Try to avoid AI autocomplete tools (GitHub Copilot Extension, Cursor) during the learning phase.
  - Save some 💰💰💰.
- Use AI as a learning assistant.
  - Ask AI to explain code.
  - Ask AI question about the code.
  - Ask AI to improve code.
- Do not copy/paste without understanding the code.