

# Important Instructions

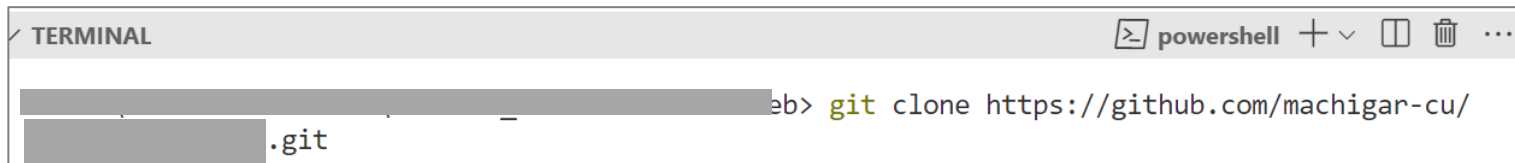
- การส่ง Assignment ผ่าน Github Classroom
  - งานใน Assignment 10 เป็นงานต่อจาก Assignment 09 แต่ใช้ Repository เริ่มต้นที่ให้มา ผ่าน Assignment Invitation Link สำหรับ Assignment 10
  - การส่ง Assignment ภายใต้ Student Identity ที่กำหนดไว้ตั้งแต่ตอนรับ Invitation ของ Assignment 01
- นิสิตทดสอบ
  - Assignment มาพร้อม Test Script สำหรับทดสอบ project ให้นิสิตทดสอบตาม instruction ในสไลด์ชุดนี้
- นิสิตส่งงานโดย
  - Deploy เว็บไซต์ที่สร้างจาก Assignment ขึ้น Vercel
  - Commit และ push ขึ้น repository ที่ได้จาก Assignment
  - นิสิตต้องใส่ link ไปยัง Vercel website ของนิสิต ใน README.md ใน Repository

# Prepare Class Environment



- เปิด Folder สำหรับเก็บหน้า React Web ที่สร้างไว้ใน Assignment 03 ด้วย Visual Studio Code
- ไปที่ Menu View > Terminal

- Clone Repo ของตนเอง พิมพ์คำสั่ง `git clone REPO_URL_FROM_GITHUB`



```
TERMINAL powershell + - [x] [x] ...
C:\Users\user> git clone https://github.com/machigar-cu/
.git
```

**Note:** ถ้ายังไม่ได้ setup git credential ในเครื่อง อาจขึ้นว่า repository not found ให้เพิ่ม user ของ git account ตามด้วย @ ไปด้านหน้า github

`git clone https://YOUR_USER@github.com/machigar-cu/.....git`

- **YOUR\_USER** คือ Github user ของนิสิต

- เปิด Folder ที่ Clone มาได้ (`a10-swdevprac2-xxxxxx`)  
ใน Visual Studio Code

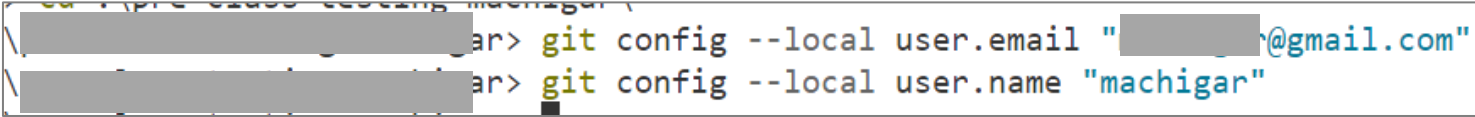
- จะพบไฟล์ตั้งต้นของ project

# Prepare Class Environment



- ไปที่เมนู **View > Terminal**
  - Configure git username และ email สำหรับ local repository

```
git config --local user.email "example@email.com"
git config --local user.name "your_userName"
```



```
machigar> git config --local user.email "example@gmail.com"
machigar> git config --local user.name "machigar"
```
  - ติดตั้ง node packages `npm install`
  - Copy ไฟล์เตอร์ `public` และ `src` จาก project ของ Assignment 09 มาทับ ไฟล์เตอร์ `public` และ `src` ใน project ของ Assignment 10 นี้
  - ลอง run project จะได้ project ที่ทำไว้ใน Assignment 09 `npm run dev`

# Prepare Class Environment



- วางไฟล์ `.env.local` ที่ให้มาสำหรับ setup NextAuth ไว้ที่ root ของ project directory
- เนื่องจากในการ deploy ขึ้น Vercel ต้องนิยาม Type ของข้อมูล รวมถึงข้อมูลชนิด session ใน Next-Auth ด้วย
  - ดาวนิโหนดไฟล์ `next-auth.d.ts` ที่ให้มา
  - วางไฟล์ไว้ที่ `src\next-auth.d.ts`
- สร้าง user ที่มี role เป็น user ธรรมดา ให้มีข้อมูลดังนี้
  - name: `Alice`
  - email: `alice@vaccinebooking.com`
  - password: `g00dH3@!th`
  - tel: `0894452632`
  - role: `user`

```
import NextAuth from 'next-auth'

declare module "next-auth" {
  interface Session {
    user: {
      _id: string,
      name: string,
      email: string,
      role: string,
      token: string
    }
  }
}
```

# Prepare Class Environment



- ใช้ Backend API ที่ setup ไว้จาก Week 09 ด้วย Source Code เดิม  
ปรับแต่ง Backend ดังต่อไปนี้ เพื่อให้รองรับการทำงานกับ Swagger Open API
- เปิดไฟล์ `server.js` ปรับให้ใช้งาน CSS จาก CDN  
<https://cdnjs.cloudflare.com/ajax/libs/swagger-ui/4.1.0/swagger-ui.min.css>
  - มีขั้นตอนการแสดงผลหน้า Swagger Open API บน Vercel อาจผิดพลาด

```
    },  
    apis:['./routes/*.js'],  
  };  
  const swaggerDocs=swaggerJsDoc(swaggerOptions);  
  const CSS_URL = "https://cdnjs.cloudflare.com/ajax/libs/swagger-ui/4.1.0/swagger-ui.min.css";  
  app.use('/api-docs',swaggerUI.serve, swaggerUI.setup(swaggerDocs, {customCssUrl:CSS_URL}));  
  
  // Handle unhandled promise rejections  
  process.on("unhandledRejection", (err, promise) => {  
    console.log(`Error: ${err.message}`);  
  });
```

# Prepare Class Environment



ปรับแต่ง Backend ดังต่อไปนี้ เพื่อให้รองรับการทำงานกับ Swagger Open API

- เปิดไฟล์ `routes\auth.js` เปิด route สำหรับให้ register user Swagger Open API ผ่าน ได้

```
58  /**
59  * @swagger
60  * /auth/register:
61  *   post:
62  *     summary: Create a new user
63  *     tags: [User]
64  *     requestBody:
65  *       required: true
66  *       content:
67  *         application/json:
68  *           schema:
69  *             $ref: '#/components/schemas/User'
70  *     responses:
71  *       201:
72  *         description: The user was successfully created
73  *         content:
74  *           application/json:
75  *             schema:
76  *               $ref: '#/components/schemas/User'
77  *       500:
78  *         description: Some server error
79  */
80  router.post("/register", register);
```

# Prepare Class Environment



- Commit การเปลี่ยนแปลงบน Backend ขึ้น github แล้วรอให้ deployment บน Vercel ได้รับการอัปเดต
- ตรวจสอบให้แน่ใจว่า **HOST** environment variable ของ Vercel Deployment **ไม่** / ด้านหลัง

<>

HOST

Development, Preview, Production

.....

Name

HOST

Value

https://vaccine-back[REDACTED].vercel.app

# Prepare Class Environment



- เปิด Swagger UI
- สร้าง user ที่มี role เป็น user ธรรมดา ให้มีข้อมูลดังนี้
  - name: Alice
  - email: alice@vaccinebooking.com
  - password: g00dH3@!th
  - tel: 0894452632
  - role: user

**POST** /auth/register Create a new user

Parameters

No parameters

Request body **required**

```
{
  "name": "Alice",
  "email": "alice@vaccinebooking.com",
  "tel": "0894452632",
  "role": "user",
  "password": "g00dH3@!th"
}
```





## Assignment 10: NextAuth & Data Posting

# NextAuth & Data Posting



ใน `vaccine-booking-app` เพิ่มการ Sign-In เป็นการ Log-In ไปยัง Backend API

- เพิ่มการใช้งาน **NextAuth** โดยใช้ NextAuth version **4.24.8** เข้าไปใน `vaccine-booking-app`
- ในโฟลเดอร์ `src\libs` ที่เก็บฟังก์ชันต่าง ๆ
  - สร้างไฟล์ `userLogIn.tsx` บรรจุ `async` ฟังก์ชัน `userLogIn` สำหรับส่ง **POST** Request ไปยัง Backend API ที่ domain ของ Backend ของนิสิต ที่ path <your-backend-domain:443/api/v1/auth/login>
    - Content-Type เป็น `application/json`
    - body เป็น JSON ในรูปแบบดังต่อไปนี้  
`userEmail` = ค่าอีเมลที่ user ป้อน  
`userPassword` = รหัสผ่านของ user
- ให้การ Sign-In ผ่าน NextAuth เป็นการเรียกฟังก์ชัน `userLogIn` นี้

```
{  
  email: userEmail,  
  password: userPassword,  
}
```



- ใน component **TopMenu** สร้าง Link ไปยังหน้า Sign-In ไว้ด้านซ้ายมือ
  - รองรับการ Sign-In ด้วย email และ password

<u>Sign-In</u>	Menu Item <b>Booking</b>	วาง logo
----------------	-----------------------------	----------

- เมื่อ Sign-In สำเร็จ Link นี้จะเปลี่ยนเป็น Link ไปยังหน้า Sign-Out แทน

# Sending Bearer Token with API Request



- ปรับแต่ง NextAuth ให้รองรับ Token ที่ส่งมาจาก Backend API หลังการ Log-In
- ในไฟล์เตอร์ `src\libs` ที่เก็บฟังก์ชันต่าง ๆ
  - สร้างไฟล์ `getUserProfile.tsx` บรรจุ `async` ฟังก์ชัน `getUserProfile` สำหรับส่ง GET Request ไปยัง Backend API พร้อมแนบ Bearer Token โดยส่งไปที่ [your-backend-domain:443/api/v1/auth/me](http://your-backend-domain:443/api/v1/auth/me)
  - ฟังก์ชัน `getUserProfile` return ค่าเป็น JSON Object ที่ได้รับมาจาก backend API
    - JSON Object ของ response จาก Backend API มีโครงสร้างดังตัวอย่าง

```
{
  "success": true,
  "data": {
    "_id": "65ed348b84ff40c82db718cc",
    "name": "Alice",
    "email": "alice@vaccinebooking.com",
    "tel": "0884411524",
    "role": "user",
    "createdAt": "2024-03-10T04:18:19.435Z",
    "__v": 0
  }
}
```

# Sending Bearer Token with API Request



- ที่หน้า Page `page.tsx` ของ route สำหรับการนัดรับวัคซีน ที่ <http://localhost:3000/booking>
  - หาก User ทำการ Log-In แล้ว ให้เรียกฟังก์ชัน `getUserProfile` เพื่อดึงข้อมูล User Profile
  - แสดงข้อมูล User Profile ไว้ที่ด้านบนของ Page
    - Name จาก `name` ของ User
    - Email จาก `email` ของ User
    - Tel. จาก `tel` ของ User
    - Member Since จาก `createdAt` ของ User

# Sending Bearer Token with API Request



- เพิ่มการเข้าถึง User Session จาก Client Component โดย
  - สร้าง client component ใน `src/ providers/NextAuthProvider.tsx` ให้ทำหน้าที่เป็น Customized Session Provider
  - ที่ `app/layout.tsx` เพิ่มการเข้าถึง Session ผ่าน `getSession()` แล้วนำไปใช้เป็น session สำหรับ `NextAuthProvider`
  - นำ `NextAuthProvider` ไปครอบ children ทั้งหมด เพื่อให้ client component ทั้งหมดเข้าถึง session ได้
- ที่ Component `Banner.tsx` บนหน้าแรก
  - หาก User ได้ทำการ Log-In แล้ว ให้แสดงข้อความที่ด้านบนขวาของ Banner ว่า `Welcome user`
    - `user` = ชื่อ (name) ของ User ที่ Log-In  
เช่น `Welcome Alice`

# Sending Bearer Token with API Request



- ทดสอบด้วยข้อมูลสำหรับ log-in ของ user account ที่มีอยู่แล้วบน Backend API ดังนี้
  - email: `alice@vaccinebooking.com`
  - password: `g00dH3@!th`

# Assignment test cases

Assignment 10 มี 3 test cases (รวม autograde 6 คะแนน)

- ลอง run test case ที่เครื่องของตนเอง ไปที่เมนู View > Terminal

- Run test case 1 (3 คะแนน) ด้วยคำสั่ง

```
npm run test -- case1.test.tsx
```

- Run test case 2 (1 คะแนน) ด้วยคำสั่ง

```
npm run test -- case2.test.tsx
```

- Run test case 3 (2 คะแนน) ด้วยคำสั่ง

```
npm run test -- case3.test.tsx
```



# Deploy to Vercel

- ใน NextAuth สำหรับ NextJs 13.4+ หากนิยาม `authOptions` ใน `route.ts` แบบดั้งเดิม จะไม่สามารถ build project ได้ จึงไม่สามารถ deploy ขึ้น Vercel ได้
  - แสดง Type Error: 'AuthOptions' is not assignable to type 'never'
- แก้ไขได้โดยแยกการประกาศ `authOptions` ไปยังไฟล์อื่นที่ไม่ใช่ `route.ts`
- สร้างไฟล์ `src\app\api\auth\[...nextauth]\authOptions.ts`
- ย้ายส่วนการประกาศ `authOptions` จาก `route.ts` มายัง `authOptions.ts`

# Deploy to Vercel

- authOptions.ts

```
import { AuthOptions } from "next-auth";
import CredentialsProvider from "next-auth/providers/credentials";
import UserLogin from "@libs/userLogin";

export const authOptions: AuthOptions = {
  providers: [
    CredentialsProvider({
      // The name to display on the sign in form (e.g. "Sign in with...")
      name: "Credentials",
      // `credentials` is used to generate a form on the sign in page.
      // You can specify which fields should be submitted, by adding keys to the `credentials` object.
      // e.g. domain, username, password, 2FA token, etc.
      // You can pass any HTML attribute to the <input> tag through the object.
      credentials: {
        email: { label: "Email", type: "email", placeholder: "email" },
        password: { label: "Password", type: "password" },
      },
      async authorize(credentials, req) {
        if (!credentials) return null;
        const user = await UserLogin(credentials.email, credentials.password);
        if (user) {
          // Any object returned will be saved in `user` property of the JWT
          return user;
        } else {
          // If you return null then an error will be displayed advising the user to check their details.
          return null;

          // You can also Reject this callback with an Error thus the user will be sent to the error page with the error message
        }
      },
    }),
  ],
  session: { strategy: "jwt" },
  callbacks: {
    async jwt({ token, user }) {
      return { ...token, ...user };
    },
    async session({ session, token, user }) {
      session.user = token as any;
      return session;
    },
  },
};
```

# Deploy to Vercel

- `route.ts`

```
src > app > api > auth > [...nextauth] > TS route.ts > ...  
1   import NextAuth from "next-auth/next";  
2   import {authOptions} from "@/app/api/auth/[...nextauth]/authOptions"  
3  
4   const handler = NextAuth(authOptions);  
5   export { handler as GET, handler as POST };|
```

- ในทุกไฟล์ที่มีการย้ายส่วนการ import `authOptions` จาก `route.ts` ด้วย

`import { authOptions } from "../api/auth/[...nextauth]/authOptions";`  
ให้แก้ไขให้ import `authOptions` จาก `authOptions.ts`

```
import { authOptions } from "../api/auth/[...nextauth]/authOptions";
```

- ทดสอบการ build ด้วยคำสั่ง `npm run build`

# Development & Deployment

- ใช้แนวทางการ Deploy แบบเดียวกับใน Assignment A03 ถึง A09
    - สร้าง Repository บน Personal GitHub Account สำหรับ Assignment A10
      - ดู Instructions ใน Assignment A01
    - Config ให้ local project มี multiple remote repositories และ Push งานขึ้น repositories ทั้งสอง
      - ดู Instructions ใน Assignment A01
    - Deploy project จาก Personal GitHub Account ขึ้น Vercel
      - ดู Instructions ใน Assignment A01
    - Configure และ Deploy บน Vercel โดยระบบจะตรวจจับได้ว่าเป็น Next.js Framework
- ใส่ link ไปยัง Vercel website ของนิสิต ใน **README.md** ใน Repository

# Deploy to Vercel

- เพื่อให้ Site ที่ Deploy ขึ้น Vercel สามารถทำงานได้แบบ Site ที่เราทดสอบที่ localhost เราต้องตั้ง **environment variables** ซึ่งประกาศไว้ที่ไฟล์ `.env.local` ให้กับ Vercel ด้วย
- ในการ Configure Project ช่วง deploy
  - เลือกส่วน **Environment Variables** กำหนด Environment Variables ต่อไปนี้
  - **NEXTAUTH\_SECRET** กำหนดค่าเป็นค่าเดียวกับที่กำหนดไว้ใน `.env.local`
  - เราจะมาใส่ **NEXTAUTH\_URL** หลังจาก Deploy สำเร็จ เพราะต้องใช้ URL ของ Vercel ที่ได้มาจากการ Deploy

```
$ .env.local X
$ .env.local
1  NEXTAUTH_URL=http://localhost:3000
2  NEXTAUTH_SECRET='Ey7nTKnggBc0bRN8WUjyShw2qzOZ6KW4fUyqcKBePxY='
```

**Configure Project**

Project Name  
2110507-a10-done

Framework Preset  
Next.js

Root Directory  
./

> Build and Output Settings

Environment Variables

Key	Value (Will Be Encrypted)
NEXTAUTH_SECRET	Bc0bRN8WUjyShw2qzOZ6KW4fUyqcKBePxY=

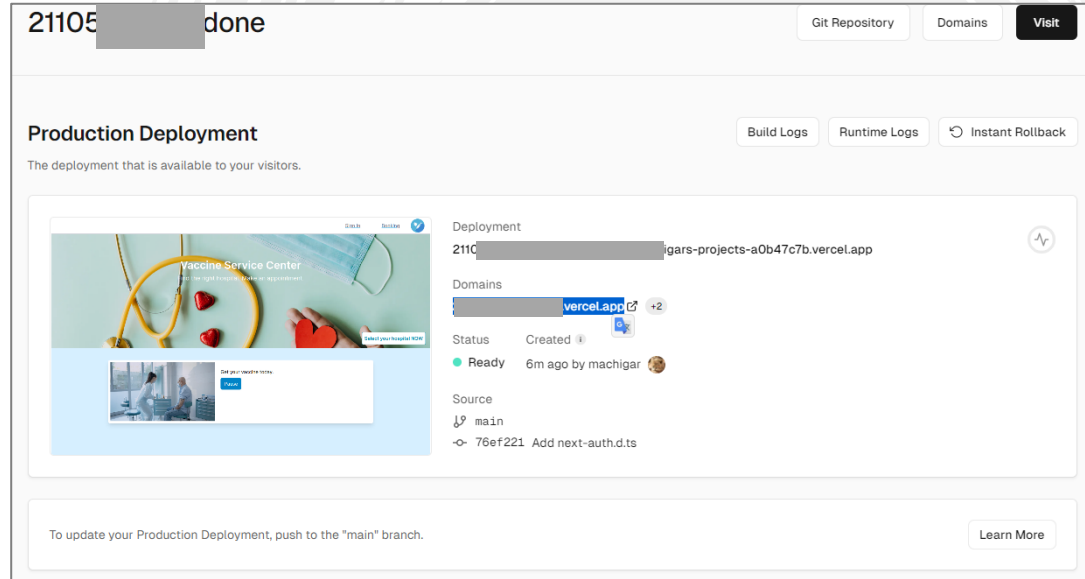
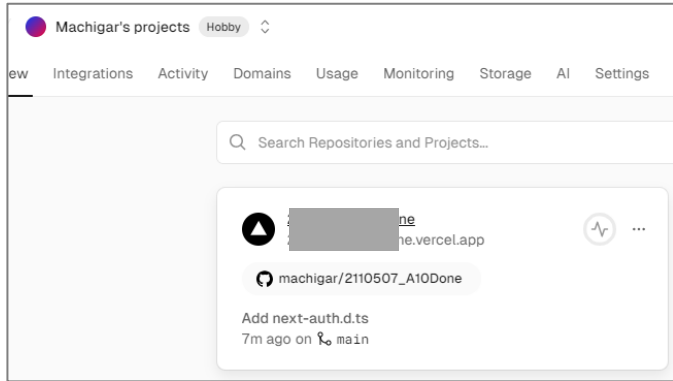
TIP: Paste a .env above to populate the form

Learn more about [Environment Variables](#)

Deploy

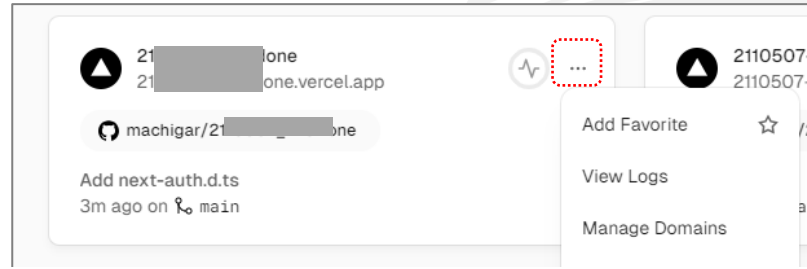
# Deploy to Vercel

- Deploy เว็บไซต์ขึ้น Vercel จนเสร็จ และ add domain
- Copy domain URL ที่ได้



# Deploy to Vercel

- ไปที่ Deployment ของเว็บไซต์
- เลือกปุ่มเมนู ... > Settings



Key

Value

NEXTAUTH\_URL|

https://a10-swdevprac2.vercel.app

+ Add Another

📄 Import .env

or paste the .env contents above

Save

Security

Advanced

☒ Production

☒ Preview [Select custom branch](#)

☒ Development

Key

Value

NEXTAUTH\_URL

https://2110507-...vercel.app/

+ Add another

📄 Import

TIP: Paste a .env above to populate the form

Save

# Deploy to Vercel

- Redeploy Project
  - ไปที่ Deployments เลือก Deployment ล่าสุด
  - เลือก Redeploy และยืนยันการ Redeploy to Production

The screenshot shows the Vercel interface for a project named 'Machigar's projects'. The 'Deployments' tab is selected and highlighted with a red dashed box. Below the tab, the 'Deployments' section shows a list of deployments. The top deployment, 'nxleoenzz', is in a 'Ready' state and is highlighted with a red dashed box. A context menu is open for this deployment, with the 'Redeploy' option highlighted by a red dashed box. The menu also includes options for 'Instant Rollback', 'Promote', and 'Inspect Deployment'.

## Redeploy to Production

You are about to create a new Deployment with the same source code as your current Deployment, but with the newest configuration from your Project Settings.

The following Domains will point to your new Deployment:

2110[redacted].vercel.app

☐ Use existing Build Cache

[Learn more about caching](#)

Cancel

Redeploy