**Міністерство освіти і науки України**

**Національний університет харчових технологій**

Кафедра інформаційних технологій, штучного інтелекту та кібербезпеки

**ЛАБОРАТОРНА РОБОТА № 2**

з дисципліни «Технології розробки та інтеграції WEB-орієнтованих систем»

на тему: «Створення full-stack web -додатка.»

Виконав: Студент I курсу  
групи КН-1-3М

Кучерявий М. В.

Перевірив:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ — 2025**

**Мета роботи**

Набуття навичок з створення додатка за допомоги TypeScript

**Хід виконання роботи**

Згідно завдання змінимо код лоабоарторної роботи 1 для коректного виконання з використанням TypeScript. Зкопіюмо в нову папку файл package.json для збереження залежностей та встановимо нові:

*npm i*

*npm install --save-dev typescript*

*npm install -D @types/express*

TSC вже був встановлений на ПК глобально тому будем використовувати його в цьому варіанті. Створимо tsconfig.json:

*tsc -init*

Налаштуємо tsc а саме внесемо такі зміни:

*"outDir": "./dist"* для отримання результау в папку dist

*"moduleResolution": "NodeNext"*

*"module": "NodeNext"*

Відредагуємо скрипти в package.json оскільки index.js файл тепер буде в папці dist:

"scripts": {

"start": "node ./dist/index.js",

"dev": "nodemon ./dist/index.js"

},

В одному терміналі запустимо tsc в режимі слідкування за змінами tsc -w. В іншому терміналі npm run dev для запуска nodemon.

Створимо файл custom.d.ts для оголошення додаткових типів, що знадобляться нам в процесі налаштування контроллера. Стовримо 3 типи:  
type Person = {

  name: string;

  phoneNumber: string;

  email: string;

  userAgent: string;

  hexcolor: string;

};

type Fruit = {

  fruit: string;

  size: string;

  color: string;

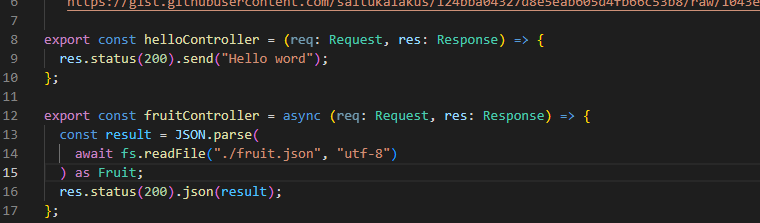
};

type personResponse = Person[];

Далі зкопіюємо файл controllers.js і змінемо розширення на .ts. Імпортуємо типи для об’єктів запиту та відповіді

import { Request, Response } from "express";

Допишемо типи в параметрах функції-котроллерах:



Також допишемо власні типи в елементах, що читається або отримується з секрвера:

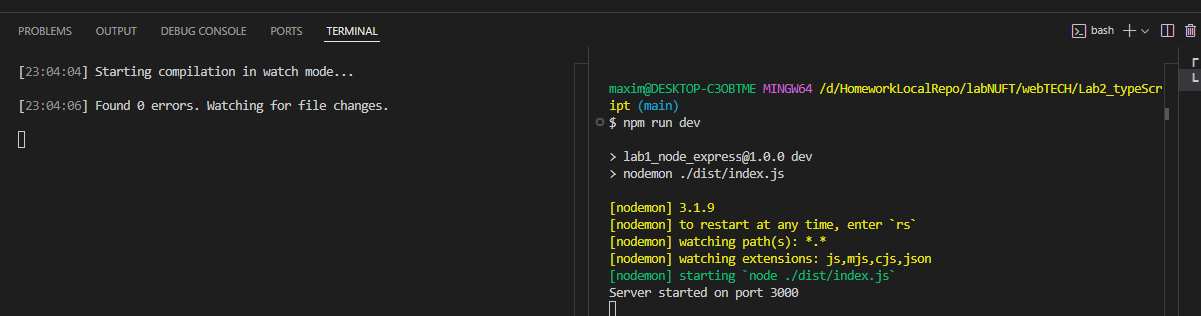


Міняємо тип помилки на any.

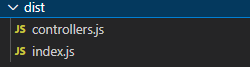
Скопіюємо та змінимо розширення на .ts файлу index.js. Загалом файл не потребує додаткової типізації крім змінної port.



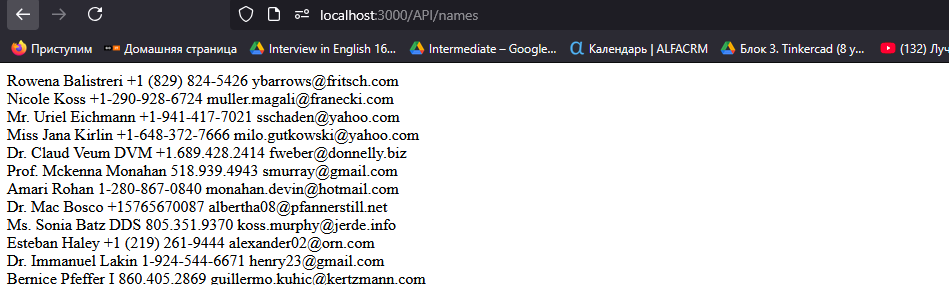
Вигляд терміналу після запуску:



Папка dist:



Протестуємо ендпоінт з іменами:



Все працює коректно.

**Відповіді на контрольні питання**

1. В чому особливість TypeScript?

По суті TypeScript є над-лаштуванням для JavaScript і допомагає вирішити проблеми динамічної типізаці. Також TypeScript має досить розгорнутий набір інструментів для реалізації OOП в його класичному розумінні з використанням класів, інтерфейсів та модифікаторів доступності.

2. Що таке Інтерфейс?

Інтерфейс це один з типів TypeScript. Дозволяє наперед задати тип та елементи певному об’єкту. Дозволяє наслідуватись з нього.

3. Що таке Анотації типів?

Це спосіб явно вказати середовищу якого типу буде певна сутність.

4. Що таке Кортеж?

Масив з фіксованим розміром та типами елементів в ньому

5. Що таке tsc?

Транспілятор TypeScript з можливістю контролю над проектом.

6. Як визначити тип об'єкт?

Можемо використати клас, інтерфейс або тип для створення об`єкту, або використати синтаксис

*const obj: {*

*a: number,*

*b: string*

*} = {*

*a: 12,*

*b: “qwe”*

*}*

7: Дайте характеристику визначенню користувацьких типів.

Користувацькі типи дозволяють спростити використання повторюваних складних типів, а також дозволяють більш зручно організувати об’єкти.

8. Як додати анотації типу?

*let a: number = 12;*

9. Що таке транспіляція?

Tранспіляція це процес перетворення одного читабельного коду на інший читабельний код. Наприклад: претворення TS на JS або перетворення JS ES6 на ES4 за допомогою Babel