Demo: Building a Node.js-Express Application Using GitHub Copilot

Prerequisites

- Node.js and npm installed
- MongoDB running locally or accessible remotely
- Git installed
- GitHub Copilot extension set up in your IDE

Lab Steps

1. Create a New Boilerplate Code for Express.js Application

Prompt: Use GitHub Copilot workspace command to generate a boilerplate code.

```
@workspace /new create a nodejs-express-app with the name books-api
```

Action: Name your project (e.g., `nodejs-express-app`) and select the template for **Express.js**.

2. Initialize Git for the Project

Command:

```
git init
```

Ask Copilot to generate a `.gitignore` file for Node.js projects. Open `Copilot edits` window and run the following prompt:

```
Generate a .gitignore for a Node.js project
```

This will create the `.gitignore` file and acceot the changes.

3. Configure MongoDB Database Connectivity

Ask Copilot to generate MongoDB connection code using Mongoose. For that open new `Copilit edits` window and run the below prompt.

```
Create a MongoDB connection configuration using mongoose for a Node.js app
```

This may create a set of files or update existing files. Accept the changes.

4. Generate a Mongoose Schema for a Book Model

Ask Copilot to generate a schema for books with the following fields:

- `title`: String (required)
- `author`: String (required)
- `publishedDate`: Date
- `pages`: Number
- `genre`: String

Open Copilot edits and run the prompt. Accept the changes suggested by Copilot.

```
Create a Mongoose schema for a book model with field validations for title,
author, publishedDate, pages, and genre
```

5. Generate API Routes for CRUD Operations

Ask Copilot to generate Express routes for CRUD operations on the book model. Use the `Copilot edits` for running below prompt.

```
Generate CRUD routes for a book model using Express.js
```