DAY 7

- *Git & GitHub*

- **Git is like a time machine for your code.**

- It is a tool that keeps a record of every version of your code, so you can always go back to a previous state if something goes wrong.

- **Install Git:** If you haven't already, download and install Git on your computer. You can get it from the official Git website: https://git-scm.com/downloads
- If you want to work with git in your project →
- Run `git init` inside the root folder of your project
- This command tells Git to start tracking changes in your project folder.

```
git status
```

- After making changes to your project (e.g., writing code), you'll want to save those changes in Git.

```
git add .
```

- The `.` means "add all changes." You can replace it with specific file names if needed.

- *gitignore*

- The `.gitignore` file is a special configuration file used in Git repositories to specify files and directories that Git should ignore.
- These ignored files and directories won't be tracked by Git or included in version control.
- **Create `.gitignore` File**

```
# Ignore node_modules directory
node_modules/

# Other entries...
```

- This saves a snapshot of your project's current state.

```
git commit -m "Initial commit"
```

- If you want to collaborate with others or back up your code online, you can create a remote repository on platforms like GitHub

- **Link Your Local and Remote Repositories**
- If you created a remote repository, you can link it to your local one

```
git remote add origin https://github.com/yourusername/hotels.git
```

- **Push Changes to Remote**
- To send your local commits to the remote repository, use the git push command

```
git push -u origin master
```

- **Pull Changes**
- If you're collaborating with others, you can fetch their changes and merge them into your code using `git pull`.

- *Host MongoDB database*

- **Now we are running locally MongoDB database.**
- All data operation is performed in a local database, so let's host our database server and make our DB online presence

- MongoDB Atlas provides a Free cluster for users where you can host your database for free.
- MongoDB Atlas offers a cloud-based platform for hosting MongoDB databases
- The free tier allows developers to explore and experiment with the database without incurring any costs.
- https://www.mongodb.com/atlas/database

- Create an account for free (I already have an account)
- Show Step-by-step Process to host MongoDB Atlas

- *Dotenv*

- The `dotenv` module in Node.js is used to manage configuration variables and sensitive information in your applications.
- It's particularly useful for keeping sensitive data like API keys, database connection strings, and other environment-specific configurations separate from your code.

```
npm install dotenv
```

- **Create a `.env` File**
- This is where you'll store your environment-specific configuration variables.
- format `VAR_NAME=value`.

```
PORT=3000
API_KEY=your-api-key
DB_CONNECTION_STRING=your-db-connection-string
```

- In your server file (usually the main entry point of your application), require and configure the `dotenv` module.

```
require('dotenv').config();
```

- **Access Configuration Variables:**

```
const port = process.env.PORT || 3000; // Use 3000 as a default
if PORT is not defined
const apiKey = process.env.API_KEY;
const dbConnectionString = process.env.DB_CONNECTION_STRING;
```

- Remember to keep your `.env` file secure and never commit it to a public version control system like Git, as it may contain sensitive information. Typically, you should include the `.env` file in your project's `.gitignore` file to prevent accidental commits.

- *Test MongoDB Cluster Postman*

- Now we can test the MongoDB Cluster and check whether our data is present or not in the online DB

- *Host NodeJS Server*

- Now we are going to host our server so that our Application or Endpoints is accessible to all the users over the Internet.
- We are using localhost and our endpoints are only accessible within our computer

- We have to make it publicly available, so there are lots of company who helps us to make our application run 24*7
- Like, AWS, Google Cloud, etc. but these charge too much amount for our application
- So we are going to use some free services to host our nodeJS application, which lots of company provides for developer purposes.
- Like, Heroku, Netlify, Render, etc