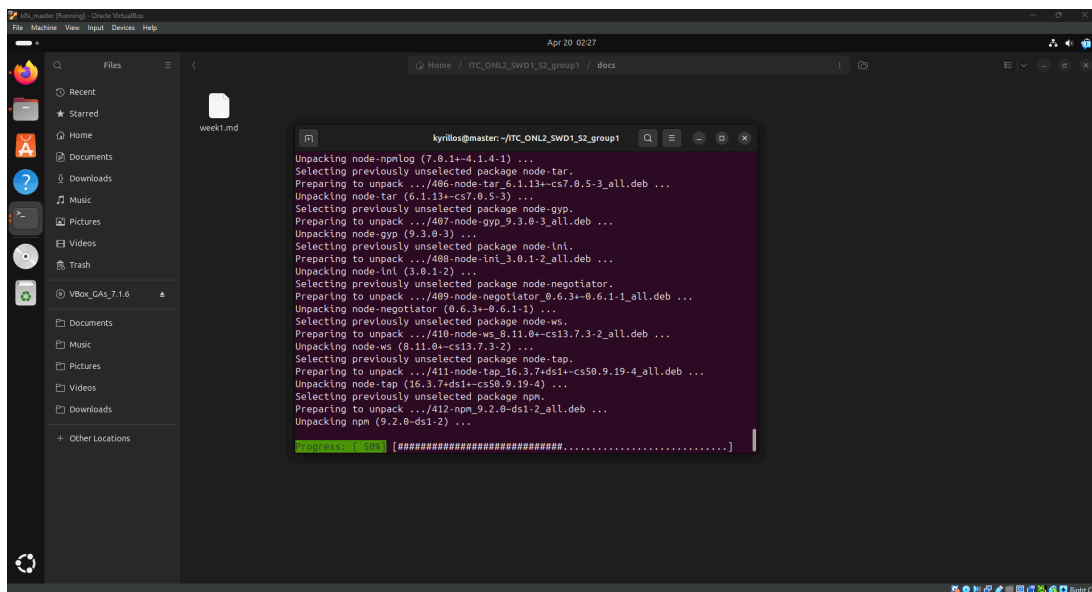
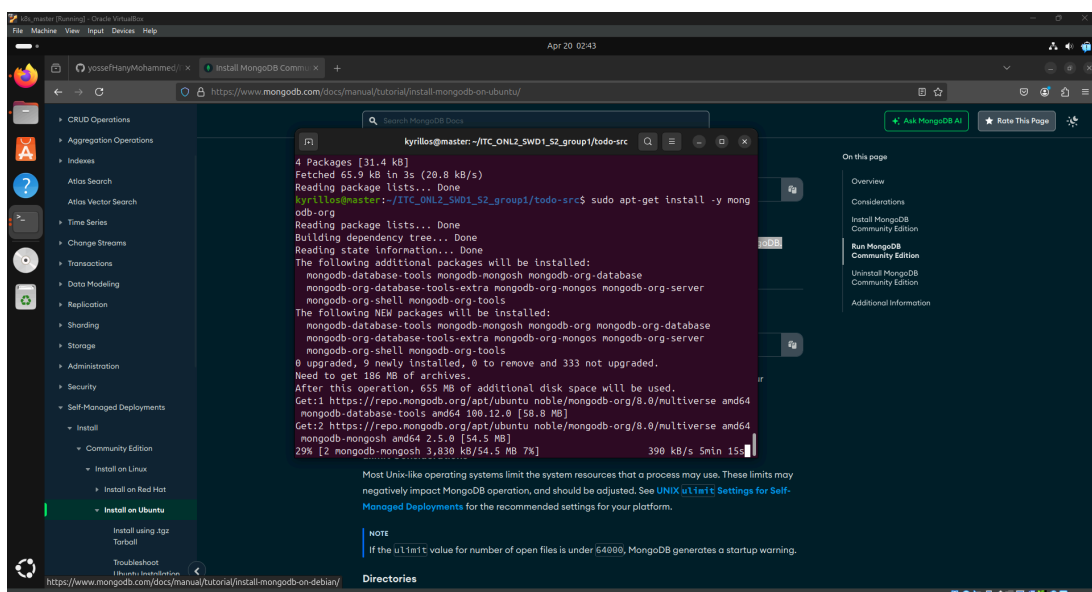


Week 1

1. install and configure devops tools
2. clone the repo
3. sudo apt update
4. install npm on machine



5. cd todo-src
6. npm install
7. trying sudo apt install -y mongodb but E: Package 'mongodb' 8. has no installation candidate
8. head to mongo official documentation for installation guide
<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>



▼ mongodb steps

Import the public key.

From a terminal, install `gnupg` and `curl` if they are not already available:

```
sudo apt-get install gnupg curl
```

To import the MongoDB public GPG key, run the following command:

```
curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \ sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \ --dearmor
```

2

Create the list file.

Create the list file

`/etc/apt/sources.list.d/mongodb-org-8.0.list` for
your version of Ubuntu.

Create the list file for Ubuntu 24.04 (Noble):

```
echo"deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
```

3

Reload the package database.

Issue the following command to reload the local package database:

```
sudo apt-get update
```

4

Install MongoDB Community Server.

You can install either the latest stable version of MongoDB or a specific version of MongoDB.

To install the latest stable version, issue the following

```
sudo apt-get install -y mongodb-org
```

9. Install production dependencies (in your repo root, where `package.json` lives):

```
npm install bcrypt chalk cors dotenv express mongoose \ connect-mongodb-session express-session validator ejs --save
```

```
# (Optional) Tidy up low-risk issues:
npm audit fix
```

10. **Install development tools:**

```
npm install nodemon eslint prettier eslint-config-prettier --save-dev
eslint --init
```

11. **Add a dev-script** to your `package.json` if it isn't there yet: **(i'm here)**

```
"dev": "nodemon -L server.js",
"start": "node server.js",
```

12. **Create a `.env`** (in the same folder as `src/index.js`) with at least:

```
MONGO_URI=mongodb://localhost:27017/todo-app
SESSION_SECRET=my_secret_key
PORT=3000
```

13. **Run MongoDB** (you've already installed it):

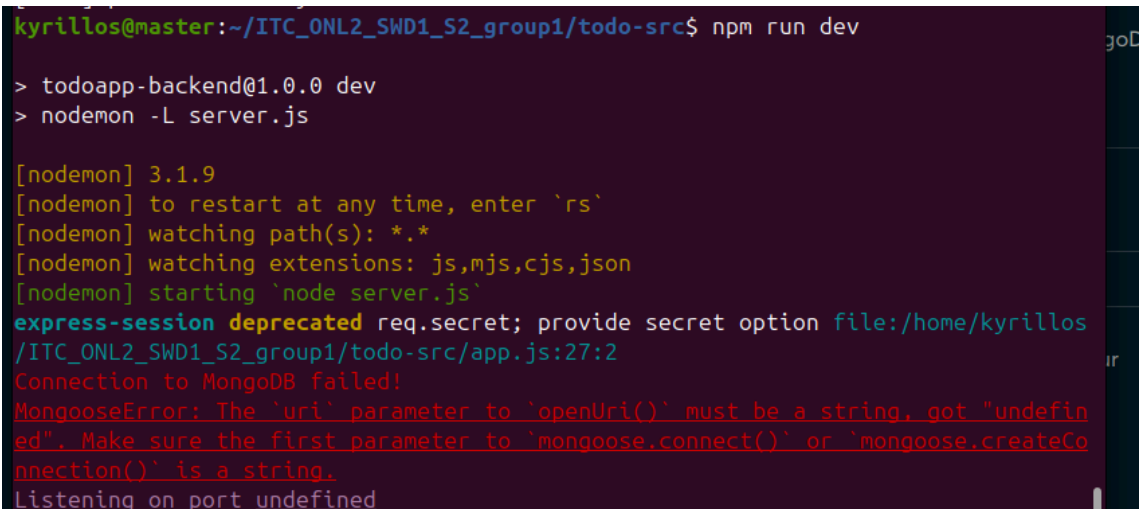
```
sudo systemctl start mongod
```

14. **Launch your app in dev mode:**

```
npm run dev
```

15. **Open `http://localhost:3000`** in your browser.

encountered this error

A terminal window with a dark background. The prompt is 'kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src\$'. The command 'npm run dev' has been executed. The output shows 'todoapp-backend@1.0.0 dev' and 'nodemon -L server.js'. Below this, nodemon version 3.1.9 is shown along with its configuration: watching path(s) *.* and extensions js, mjs, cjs, json. It then says 'starting node server.js'. An 'express-session deprecated' warning follows, indicating that req.secret is deprecated and a secret option file should be provided. Then, a 'Connection to MongoDB failed!' error occurs. The full error message is: 'MongooseError: The `uri` parameter to `openUri()` must be a string, got "undefined". Make sure the first parameter to `mongoose.connect()` or `mongoose.createConnection()` is a string.' The terminal ends with 'Listening on port undefined'.

```
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$ npm run dev
> todoapp-backend@1.0.0 dev
> nodemon -L server.js

[nodemon] 3.1.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
express-session deprecated req.secret; provide secret option file:/home/kyrillos/ITC_ONL2_SWd1_S2_group1/todo-src/app.js:27:2
Connection to MongoDB failed!
MongooseError: The `uri` parameter to `openUri()` must be a string, got "undefined". Make sure the first parameter to `mongoose.connect()` or `mongoose.createConnection()` is a string.
Listening on port undefined
```