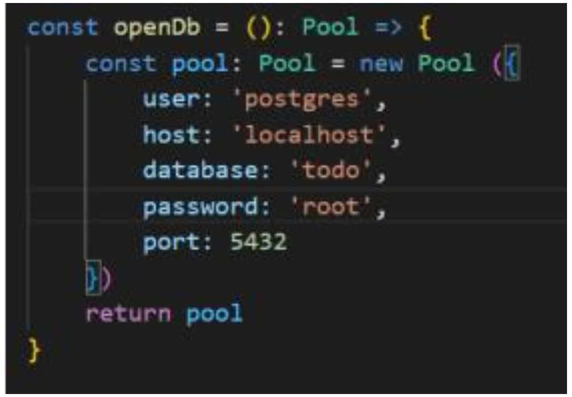
Environment Variables

Web Programming Project / Meija Lohiniva, OAMK 2023

Question 1: What challenges do we have with given example?

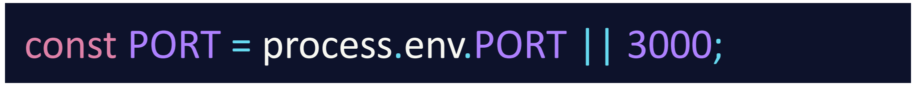
- Below is a screenshot of the example code for ToDo database connection. What challenges do we have in this code?



```
const openDb = (): Pool => {
    const pool: Pool = new Pool ({
        user: 'postgres',
        host: 'localhost',
        database: 'todo',
        password: 'root',
        port: 5432
    })
    return pool
}
```

Environment Variables

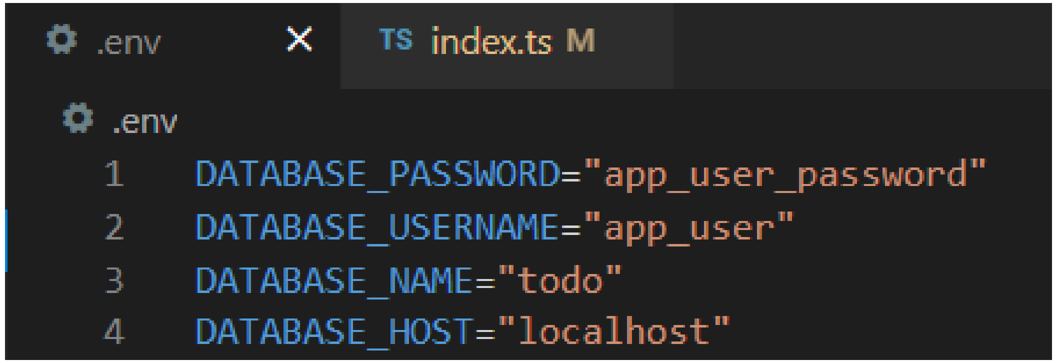
- Environment variables are a simple way to configure your Node.js app.
- Many cloud hosts (Heroku, Render, Azure etc.) use environment variables.
- Access to existing environment variables is established automatically in `process.env` object.
- `process.env` object is like any other object -> you can set and override values.



```
const PORT = process.env.PORT || 3000;
```

Environment Variables

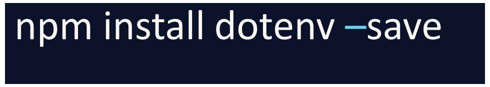
- Easy way to create project specific configuration is by using `.env` files.
- Usually you don't want to version control env files -> add it to `.gitignore`.
  - It's good practice to add a `.env.example` file in version control so it's clear what needs to be in the `.env` file.



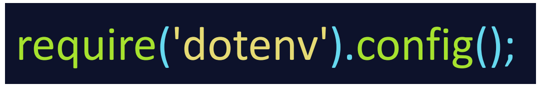
```
DATABASE_PASSWORD="app_user_password"
DATABASE_USERNAME="app_user"
DATABASE_NAME="todo"
DATABASE_HOST="localhost"
```

Environment Variables

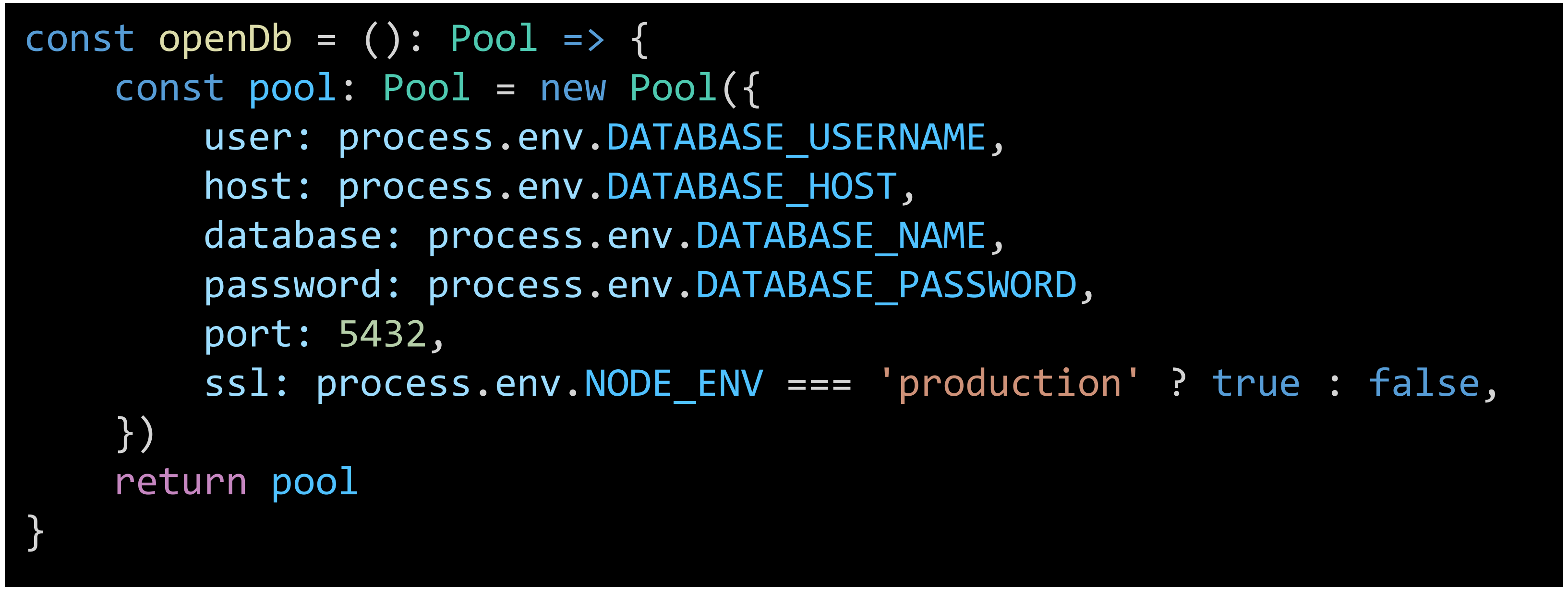
- The easiest way to load values from `.env` file is with the help of `dotenv` node module.



```
npm install dotenv –save
```



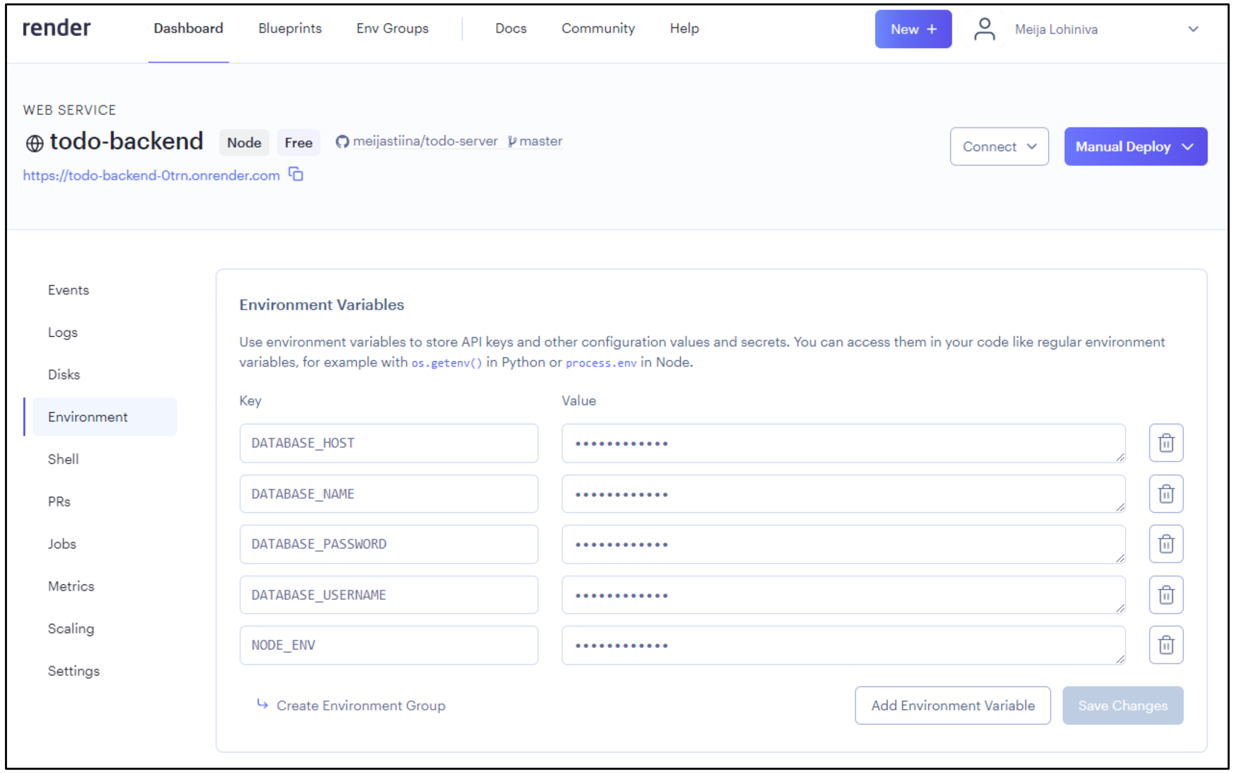
```
require('dotenv').config();
```



```
const openDb = (): Pool => {
    const pool: Pool = new Pool({
        user: process.env.DATABASE_USERNAME,
        host: process.env.DATABASE_HOST,
        database: process.env.DATABASE_NAME,
        password: process.env.DATABASE_PASSWORD,
        port: 5432,
        ssl: process.env.NODE_ENV === 'production' ? true : false,
    })
    return pool
}
```

Environment Variables

- In your cloud host you can define different credentials.
- In Render you can find environment variables under a deployed services settings.
- To determine whether app is running on development or production, you can use `NODE_ENV=production`
  - By default Node.js assumes it's running in development.



Environment Variables

Use environment variables to store API keys and other configuration values and secrets. You can access them in your code like regular environment variables, for example with `os.getenv()` in Python or `process.env` in Node.

Key	Value
DATABASE_HOST	•••••••••••••
DATABASE_NAME	•••••••••••••
DATABASE_PASSWORD	•••••••••••••
DATABASE_USERNAME	•••••••••••••
NODE_ENV	•••••••••••••

Task 1
Create simple example

1. Create a new project.
2. Install `dotenv`.
3. Create a js file.
4. Add `console.log` that prints out value of `process.env.foo`.
5. Require dotenv config.
6. Print out value of `process.env.foo` again.
7. Create `.env` file with a value for `foo`.
8. Run your code again.

Task 2 .gitignore

1. Initialize a local repository.
2. See what files are to be version controlled.
3. Create `.gitignore` file.
4. Add `.env` file into your `.gitignore`.
5. See what files are to be version controlled.

Group Work 1
Modify your app

Modify your app so that:

- Credentials are not version controlled.
- You can use different databases for development and production.

OAMK OULUN AMMATTIKORKEAKOULU