
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
Faculty of Engineering, Built Environment & IT
University of Pretoria

# COS301 - Software Engineering

# Office Booker

Team name: Kryptos Kode



| Name | Student Number | Email |
|---|---|---|
| Ying Hao Li* | u20460687 | u20460687@tuks.co.za |
| Arul Agrawal | u18053239 | u18053239@tuks.co.za |
| Grant Bursnall | u15223893 | u15223893@tuks.co.za |
| Damian Vermeulen | u20538945 | u20538945@tuks.co.za |
| Brett du Plessis | u19037717 | u19037717@tuks.co.za |

* - Team Leader

# Contents

# 1 Introduction

Office Booker is a system that allows a user to interactively map out an office space and then book conference rooms and desks within the mapped-out office space. Many employees began working from home when the COVID-19 pandemic hit and thus a system is needed to help with organising who uses the limited office space and this is done through booking the desired space for the desired time.

# 2 Github

The GitHub Repository contains all code for the Office Booker to run. The repository has is made up of multiple coding languages and contains a read Me of the contributors. Link to repo: Office Booker

## 2.1 Prerequisites

- Git Clone: `https://github.com/COS301-SE-2022/Office-Booker`
- Yarn 1.22.19
- Angular 14.0.7

## 2.2 Installation

Run the code

- *yarn install*

## 2.3 Deployment

Run the code

- *yarn start*

# 3 Frontend

User interface for the Office Booker web application.

## 3.1 Prerequisites

- Git Clone: `https://github.com/COS301-SE-2022/Office-Booker`
- Yarn 1.22.19
- Angular 14.0.7

## 3.2 Installation

The frontend will be installed with the GitHub when the repository clone has been set up appropriately.

## 3.3 Deployment

Run the following code:

- *yarn start*
- Go to your chosen browser and type http://127.0.0.1:4200/ to run the web page on local host.

# 4 Database

The Database will store all relevant data needed for the application to run.

## 4.1 Prerequisites

- Prisma Studio 3.14.0
- Docker 2.2.1

## 4.2 Installation

Run the following code in terminal in order:

- *Docker Compose up*
- *prisma migrate*
- *prisma migrate dev*
- *prisma migrate reset*

### 4.3 Deployment

- *docker compose up*
- *prisma studio*

## 5 API

The API will be used to get and send data to the database.

### 5.1 Prerequisites

- Nestjs

### 5.2 Installation

The installation is part of the yarn install packages.

### 5.3 Deployment

Run the following code:

- *yarn nx run api:serve*

# 6 Hosting

The Office Booker system is made up of three main components. These are the frontend, backend and database.

The hosting strategy uses docker containers for easy deployments. All docker containers mentioned are running on a free tier t3.micro (1 CPU and 512MB RAM) instance of AWS EC2 as specified by the system constraints.

The database is Postgresql and was chosen for ease of use. Version 14 of Postgresql is running.

The angular frontend got build to an html+css+js package, which is then hosted by nginx in a container. The frontend is accessible at https://officebooker.co.za

The nestjs backend is run inside a NodeJS LTS container and is hosted at https://api.officebooker.co.za/

For the purpose of the demo, Caddy is used as a reverse proxy to enable HTTPS and domain routing.

The backend container expects these environment variables:

- DATABASE_URL - A database url for prisma
  - `https://www.prisma.io/docs/reference/database-reference/connection-urls`
- COGNITO_USER_POOL_ID - Your AWS Cognito user pool id
- COGNITO_CLIENT_ID - Your AWS Cognito client id
- COGNITO_REGION - Your AWS Cognito region
- MAIL_HOST - SMTP Mail Host
- MAIL_USER - SMTP User
- MAIL_PASSWORD - SMTP Password
- MAIL_FROM - The Mail From Address
- MAIL_TRANSPORT - The URL
  - smtp://${MAIL_USER}:${MAIL_PASSWORD}@${MAIL_HOST}
  - filled with the above values.

Building the docker containers:

When running in production, the api url needs to specified. The api url is specified in apps/office-booker/src/environments/environment.prod.ts

First, we need nx to build our project:

Run while in the root of the project: yarn nx run-many –target=build –projects=api,office-booker –parallel

Then, to build the frontend: docker build -f ./apps/office-booker/Dockerfile . -t frontend

And the backend: docker build -f ./apps/api/Dockerfile . -t backend

Now the two containers are available on your system. A sample docker-compose is given.