

CaPPSM Deployment & Operations Guide

UMGC Capstone Project Proposal Management System (CaPPMS) Deployment & Operations Guide Version 2.0

Prepared By:

Bereket Tamrat

Ephrem Kefle

Kathryn Stewart

Marc Bueno

Tarun Lava

Yonas Mekete

Installation & User Guide Approvals

Name	Signature	Date
Professor Dr. Mir Assadullah		
Project Manager Kathryn Stewart		

Revision History

Date	Version	Description
10/20/2020	1.0	Initial Deployment & Operations Guide
11/3/2020	2.0	Included the no warranty statement

Table of Contents

Table of Contents 4

1. Introduction 6

1.1. Scope..... 6

1.2. References..... 6

1.3. Definitions, Acronyms and Abbreviations 6

2. Build..... 7

2.1. Front End 7

2.2. Back-End..... 11

2.3. File Structure..... 14

3. Installation..... 15

4. Deployment 16

5. Operations 17

5.1. Start or Restart 17

5.2. Stop 17

5.3. Supervision 17

5.4. Error Messages..... 17

5.5. Frequently Asked Questions 17

6. License 18

List of Figures

Figure 1 npm Included in node.js Installation	8
Figure 2 node.js Version Command	8
Figure 3 npm Version Command.....	8
Figure 4 Angular Install Command	9
Figure 5 Angular Version Command.....	9
Figure 6 Angular Version Output	9
Figure 7 Add to PATH.....	10
Figure 8 Create Directory	10
Figure 9 Code . Command.....	10
Figure 10 Front-End Start Command	11
Figure 11 Front-End Start Output	11
Figure 12 Select PostgreSQL Components.....	12
Figure 13 Create New Database	12
Figure 14 Name the New Database	12
Figure 15 Query to Insert FAQ.....	13
Figure 16 Application.Properties File Location.....	13
Figure 17 Application.Properties File Contents.....	13
Figure 18 Front-End File Structure.....	14
Figure 19 Back-End File Structure	15
Figure 20 Front-End Start Command	17
Figure 21 Front-End Start Output	17

List of Tables

Table 1 Reference Documents	6
Table 2 Acronyms.....	6

1. Introduction

The University of Maryland Global Campus (UMGC) Capstone Project Proposal Management System (CaPPMS) is designed to assist the professor with collecting, reviewing, updating, and exporting project ideas for the capstone class. This web-based application will be used by external entities to submit detailed proposals of projects. The professor will export these details for the UMGC Software Engineering (SWEN) 670 students to select as their project for the semester.

1.1. Scope

This deployment and operations guide provides the technical details and steps necessary to setup and run the UMGC Capstone Project Proposal Management System (CaPPMS). This document covers the steps necessary to install, start, stop, restart, supervise, and debug the CaPPMS system. It is intended for IT and Operations personnel with various degrees of familiarity with the software. The target user should be familiar with the following concepts: Postgres, SpringBoot, Git, GitHub, Azure and Docker.

1.2. References

Table 1 Reference Documents

Document	Version
CaPPMS Software Requirement Specification	2.0
CaPPMS Project Plan	2.0
CaPPMS Test Plan	1.0
CaPPMS Requirements Matrix	1.0

1.3. Definitions, Acronyms and Abbreviations

Below are the terms and abbreviations used in this document.

Table 2 Acronyms

Acronym	Definition
API	Application Programming Interface
CaPPMS	Capstone Project Proposal Management System

Acronym	Definition
IDE	Integrated Development Environment
PM	Project Manager
RDBMS	Relational Database Management Systems
REST	Representational State Transfer
SDLC	Software Development Life Cycle
SRS	Software Requirement Specification
SWEN	Software Engineering
UMGC	University of Maryland Global Campus
WBS	Work Breakdown Structure

2. Build

2.1. Front End

The CaPPMS application was built using Angular 10 as a front-end, Spring boot 2 RESTful API as a backend, connected to PostgreS via a hibernate JPA Provider.

The tools and technologies used to build the front-end are:

- Integrated Development Environment (IDE) -Visual Studio Code - Angular App development
- Angular 10
- Angular CLI
- NodeJS and NPM
- Bootstrap 4
- HTML5

To configure the front-end code on a local machine for building and development, complete the following steps. These steps assume a Windows development environment.

1. Install node.js
 - a. Navigate to <https://nodejs.org/en/>
 - b. Press download for the version 'Recommended for most users'

- c. Save the install file
- d. Open the install file and follow the instructions
- e. Note that npm is selected for installation as part of node.js.

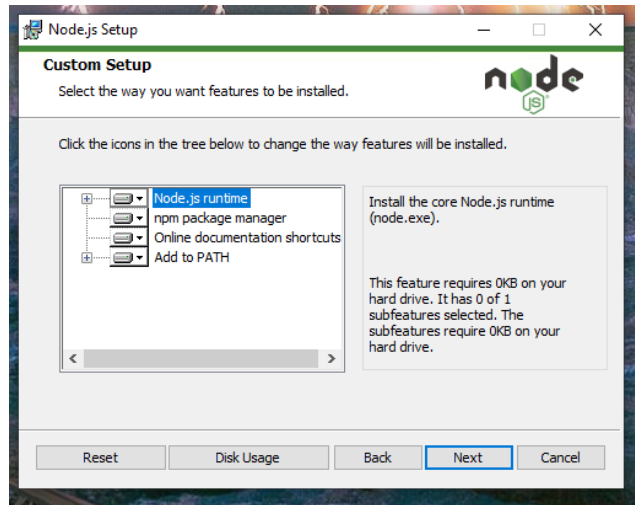


Figure 1 npm Included in node.js Installation

2. Open a command prompt and confirm node.js installation
 - a. > node -v

```
C:\Users\kathr>node -v
v12.19.0
```

Figure 2 node.js Version Command

3. From the open command prompt confirm npm installation
 - a. > npm -v

```
C:\Users\kathr>npm -v
6.14.8
```

Figure 3 npm Version Command

4. From the open command prompt install angular/cli
 - a. > npm install -g @angular/cli

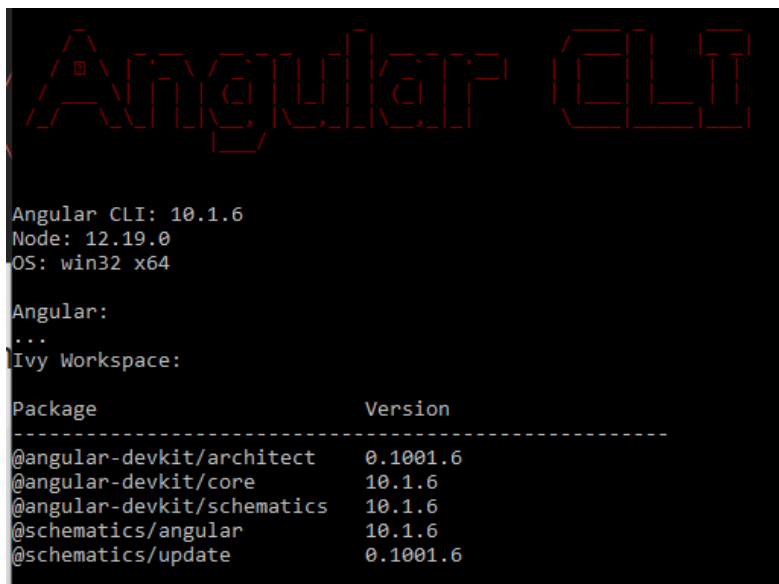

```
C:\Users\kathr>npm install -g @angular/cli
```

Figure 4 Angular Install Command

5. From the open command prompt confirm angular cli installation. Additional information regarding these commands can be found here: <https://cli.angular.io/>
 - a. > ng --version

```
C:\Users\kathr>ng --version
```

Figure 5 Angular Version Command



```
Angular CLI

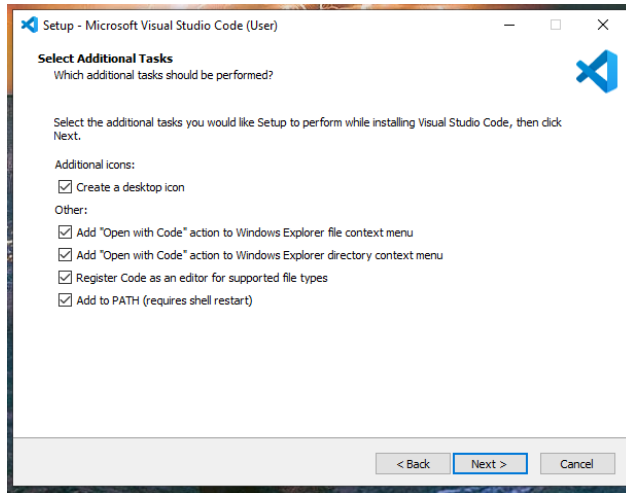
Angular CLI: 10.1.6
Node: 12.19.0
OS: win32 x64

Angular:
...
Ivy Workspace:

Package                           Version
-----
@angular-devkit/architect         0.1001.6
@angular-devkit/core              10.1.6
@angular-devkit/schematics        10.1.6
@schematics/angular              10.1.6
@schematics/update                0.1001.6
```

Figure 6 Angular Version Output

6. Install VS Code
 - a. Navigate to <https://code.visualstudio.com/download>
 - b. Press the button to download the installer for Windows
 - c. Save the install file
 - d. Open the install file and follow the instructions
 - e. Ensure the Add to PATH check box is selected, plus any other features desired by the developer.

*Figure 7 Add to PATH*

7. Create a folder to store the project and navigate to that folder
 - a. `> mkdir CaPPMS_FrontEnd`
 - b. `> cd CaPPMS_FrontEnd`

```
C:\Users\kathr\Documents>mkdir CaPPMS_FrontEnd
C:\Users\kathr\Documents>cd CaPPMS_FrontEnd
C:\Users\kathr\Documents\CaPPMS_FrontEnd>
```

Figure 8 Create Directory

8. Flag the directory to indicate its use for code
 - a. `> code .`

```
C:\Users\kathr\Documents\CaPPMS_FrontEnd>code .
```

Figure 9 Code . Command

9. Open VS Code and the folder that was just created
 - a. File -> Open Folder
10. Connect to GitHub and clone the repository to this folder.
11. To run the front-end open a terminal window and run this command:
 - a. `> ng serve`

```
C:\Users\kathr\Desktop\Project Tracker Frontend\Project Tracker Frontend\project-tracker>ng serve
```

Figure 10 Front-End Start Command

```
chunk {main} main.js, main.js.map (main) 150 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 1.06 MB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 5.05 MB [initial] [rendered]
Date: 2020-10-16T18:28:14.947Z - Hash: a2c1f8a2af6a14bdec7c - Time: 14678ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

Figure 11 Front-End Start Output

12. Navigate to <http://localhost:4200/> to view the program.

2.2. Back-End

Server-side tools and technologies include:

- IDE – Eclipse - Spring boot API development
- Maven 3.3
- Spring Boot 2.2.1
- Spring Framework 5.2
- JDK 1.8
- Spring Data JPA (Hibernate)
- Embedded Tomcat 8.5
- Postgres v12 Database

To configure the back-end code on a local machine for building and development, complete the following steps.

1. Install PostgreSQL
 - a. Download from <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>
 - b. Ensure that all the options are selected for installation

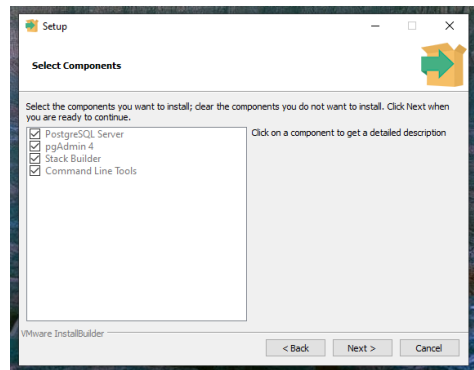


Figure 12 Select PostgreSQL Components

- c. Open pgAdmin 4
- d. Create a database called ProjectTracker

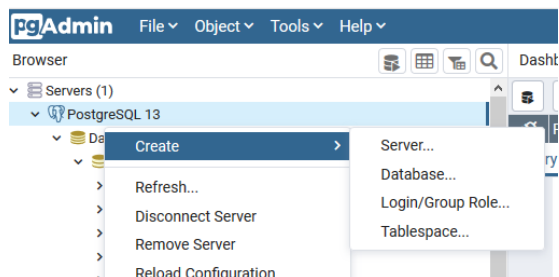


Figure 13 Create New Database

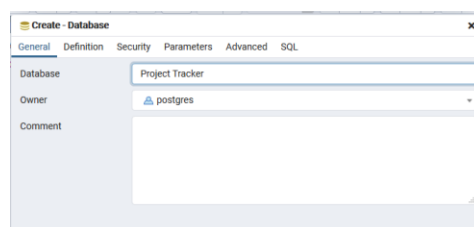


Figure 14 Name the New Database

- e. Enter the FAQ by right clicking on the FAQ table, selecting Scripts, and INSERT Script. Repeat as many times as needed to enter all the FAQs.

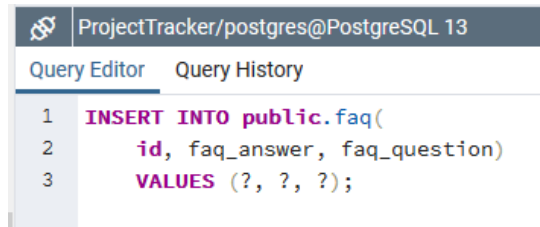


Figure 15 Query to Insert FAQ

2. Using the IDE of choice (i.e. eclipse) import the code from GitHub
3. Open the application.properties folder and update the username and password for the newly created database

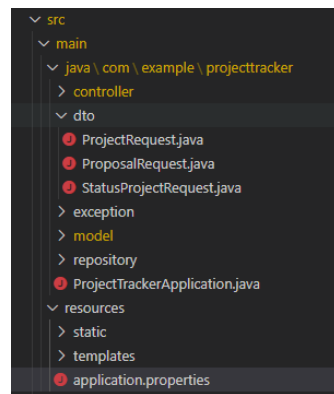


Figure 16 Application.Properties File Location

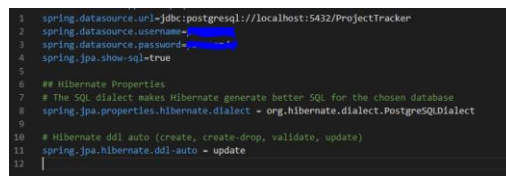


Figure 17 Application.Properties File Contents

4. Right click the project and Build
5. Right click and run the project

2.3. File Structure

There are two separate repositories of code, one for the front-end and one for the back-end.

The front-end code is contained in an angular project with the following structure:

about
create-project
faq
home
login
newadmin
project-details
project-list
project-request
proposal-table
update-project
app-routing.module.ts
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts
faq.service.spec.ts
faq.service.ts
faq.spec.ts
faq.ts
project-request.spec.ts
project-request.ts
project.service.spec.ts
project.service.ts
project.spec.ts
project.ts
user.spec.ts
user.ts

Figure 18 Front-End File Structure

The back-end code is contained in a java project with the following structure:

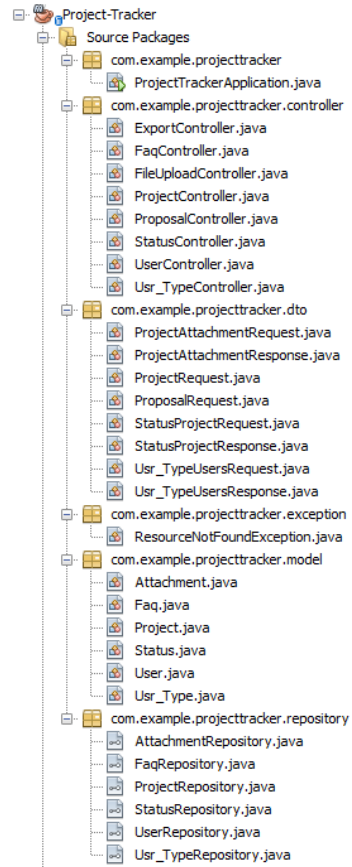


Figure 19 Back-End File Structure

3. Installation

The front and backend software will be installed on two dockers. The following specifications are needed for the docker:

- Front End:

- Windows 10
- Microsoft Word
- node.js
-
- angular/cli
- Back End:
 - Maven
 - PostgreS
 - Hibernate

Commented [MB1]: I guess node.js already contains npm (that was on the the reasons for installing node.js. IS it correct ?

4. Deployment

The CaPPMS application will be deployed in a Docker container. The DevOps team will pull the latest code from the master branch in GitHub.

The DevOps team will run the following command to deploy the code to a Docker instance running on Azure:

- make deploy

The DevOps team has created a build and make file that will pull the latest code, build the program, create the database and configure for persistent storage, deploy and start the application. This build and make file will dynamically create the domain names and ports that the website will use for external access. This file will also ensure the correct database user and password will be used by the program.

To configure the development Docker environment, the user must have Docker installed and a student account on Azure. From the application code command line type the same command and follow the prompts to connect and deploy.

To stop and close the Docker use the command

- make stop-deploy

5. Operations

5.1. Start or Restart

To run the front-end open a terminal window and run this command:

- `ng serve`

```
C:\Users\kathr\Desktop\Project Tracker Frontend\Project Tracker Frontend\project-tracker>ng serve
```

Figure 20 Front-End Start Command

```
chunk {main} main.js, main.js.map (main) 150 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 1.06 MB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 5.05 MB [initial] [rendered]
Date: 2020-10-16T18:28:14.947Z - Hash: a2c1f8a2af6a14bdec7c - Time: 14678ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

Figure 21 Front-End Start Output

5.2. Stop

To stop the program press Control + C twice to get the prompt to terminate the batch job.

Press Y at the prompt and the program will stop running.

5.3. Supervision

The CaPPMS does not require any administrative tasks to be performed while it is running.

5.4. Error Messages

The CaPPMS system will display error messages when required data is not entered by the user or the professor.

5.5. Frequently Asked Questions

1. Do I need an account to read about this application?
 - a. No. Anyone who wants to read about the application can do so in the about page.
2. Is there a way to ask for support?
 - a. The current development team will provide the professor with all documentation and code related to this project. The point of contact is the Software Engineering department of the University of Maryland Global Campus.
3. What type of database maintenance is required in the CaPPMS?

- a. PostgreSQL requires minimum maintenance and although a dedicated DBA is not required for CaPPMS, a skilled database person may be need from time to time to perform basic tasks. The following lists some recommendations to keep the database in good shape:
 - i. A backup / recovery plan should be developed. PostgreSQL offers two main approaches to backup, to address different needs, namely SQL Dumps and PITR (Point in time recovery). At the time of this writing and based on requirement gathering, the CaPPMS database should be change constantly in which case a PITR backup / restore approach would be prescribed. Therefore, the CaPPMS development team PostgreSQL dump approach be used as a backup / recovery policy. SQL Dumps allows individual tables and database objects to be recovered more efficiently in this case. SQL dumps and PITR approaches allows to leverage the benefits of incremental backups that are part of PostgreSQL. Should a drastic increase in the volume of data changed happen, a PITR strategy may be put in place.
 - ii. An assessment of the database growth rate should be done frequently by checking tablespaces growth, indexes, log files, system statistics and overall database and SQL performance tuning.
- 4. What are the steps to upgrade the CaPPMS architecture?
 - a. Perform a full database backup

6. License

The CaPPMS does not require any licenses.

This software is free to use by anyone. It comes with no warranties and is provided solely "AS-IS". It may contain significant bugs, or may not even perform the intended tasks, or fail to be fit for any purpose. University of Maryland is not responsible for any shortcomings and the user is solely responsible for the use.