



# **Technical Installation Manual**

29.09.2024

|                                    |          |
|------------------------------------|----------|
| <b>Introduction</b>                | <b>2</b> |
| Front-end                          | 2        |
| Back-end                           | 2        |
| Overview                           | 3        |
| <b>Prerequisites</b>               | <b>3</b> |
| <b>Installing Prerequisites</b>    | <b>3</b> |
| Node.js and npm                    | 3        |
| Angular CLI                        | 3        |
| AWS CLI                            | 3        |
| Git                                | 4        |
| Visual Studio Code                 | 4        |
| <b>Installation</b>                | <b>4</b> |
| Cloning the Repository             | 4        |
| Installing Dependencies            | 4        |
| Configuring AWS Amplify            | 5        |
| <b>Deployment/Running</b>          | <b>5</b> |
| Local Development                  | 5        |
| Production Deployment              | 6        |
| Accessing the Deployed Application | 6        |
| <b>Testing</b>                     | <b>6</b> |
| <b>Troubleshooting</b>             | <b>6</b> |
| <b>User Manual</b>                 | <b>7</b> |
| <b>Conclusion</b>                  | <b>7</b> |

# Introduction

Smart Inventory is a comprehensive web-based application designed to revolutionize inventory management for businesses of all sizes, including restaurants and laboratories. The system consists of two main components:

## Front-end

An Angular-based user interface that provides an intuitive and responsive experience for managing inventory.

## Back-end

A serverless architecture powered by AWS services, including DynamoDB for data storage, Lambda for serverless computing, and API Gateway for RESTful API management.

## Overview

This manual will guide you through the process of setting up both the front-end and back-end components of the Smart Inventory system on your local machine for development purposes. It will also cover the deployment process using AWS Amplify.

# Prerequisites

Before beginning the installation process, ensure you have the following software installed on your system:

- Node.js (version 18.0.0 or later)
- npm (version 10.8.1 or later)
- Angular CLI (version 18.0.2)
- AWS CLI (version 2.x)
- Git (latest version)
- Visual Studio Code (or any preferred code editor)

# Installing Prerequisites

## Node.js and npm

1. Visit the official Node.js website: \url{https://nodejs.org}
2. Download and install the latest LTS version (18.x.x) for your operating system.
3. Verify the installation by running:
  - a. `node --version`
  - b. `npm --version`

## Angular CLI

1. Install Angular CLI globally using npm:
  - a. `npm install -g @angular/cli@18.0.2`
2. Verify the installation:
  - a. `ng version`

## AWS CLI

Follow the installation guide at:

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

After installation, configure AWS CLI:

1. `aws configure`
2. Enter your AWS access key ID and secret access key when prompted.

## Git

Download and install Git from:

<https://git-scm.com/downloads>

Verify the installation:

- `git --version`

## Visual Studio Code

Download and install from:

<https://code.visualstudio.com/>

Install recommended extensions for Angular development:

- Angular Language Service
- ESLint
- Prettier

# Installation

## Cloning the Repository

1. Open your terminal or command prompt.
2. Navigate to the directory where you want to clone the project.
3. Run the following command:
  - a. `git clone https://github.com/yourusername/Smart-Inventory.git`
  - b. `cd Smart-Inventory`

## Installing Dependencies

In the project root directory, run:

- `npm install`

This will install all necessary dependencies listed in the package.json file, including:

- `@angular/animations: 18.0.0`
- `@angular/cdk:18.0.3`
- `@angular/common: 18.0.0`
- `@angular/compiler: 18.0.0`
- `@angular/core: 18.0.0`
- `@angular/forms: 18.0.0`
- `@angular/material: 18.0.3`
- `@angular/platform-browser: 18.0.0`
- `@angular/platform-browser-dynamic: 18.0.0`
- `@angular/router:18.0.0`
- `@aws-amplify/ui-angular: 5.0.16`
- `Aws-amplify: 6.3.4`
- `rxjs: 7.8.0`
- `zone.js: 0.14.3`

## Configuring AWS Amplify

1. Install the Amplify CLI globally:
  - a. `npm install -g @aws-amplify/cli@6.x`
2. Initialize Amplify in your project:
  - a. `amplify init`

1. Add necessary Amplify categories (API, Auth, Storage):
  - a. `amplify add api`
  - b. `amplify add auth`
  - c. `amplify add storage`
2. the prompts for each category to set up your backend resources.
  - a. `amplify push`

# Deployment/Running

## Local Development

To run the application locally:

1. `ng serve`
2. Open a web browser and navigate to <http://localhost:4200>

## Production Deployment

To deploy the application to AWS:

1. Build the production version of your app:
  - a. `ng build --prod`
2. Deploy using Amplify:
  - a. `amplify publish`

This will deploy your front-end to AWS Amplify hosting and update your backend resources if necessary.

## Accessing the Deployed Application

After successful deployment, Amplify will provide a URL where your application is hosted. You can access your application through this URL.

<https://main.d20fi26o9migpb.amplifyapp.com>

# Testing

To run the Cypress tests:

- Component tests:
  - `npm run test:component`

- E2E tests (make sure your app is running locally first):
  - `npx cypress run`

## Troubleshooting

If you encounter any issues during installation or deployment, please refer to the following resources:

- Angular documentation:
  - <https://angular.io/docs>
- AWS Amplify documentation:
  - <https://docs.amplify.aws/>
- Project GitHub Issues page:
  - <https://github.com/COS301-SE-2024/Smart-Inventory/issues>

If you can't find a solution, please open a new issue on the GitHub repository with a detailed description of the problem and steps to reproduce it.

## User Manual

For detailed instructions on how to use the Smart Inventory system, please refer to the user manual in the app or on the github

## Conclusion

Always ensure you're using the correct versions of packages and dependencies as specified in the package.json file to maintain compatibility and stability of the system post-development. Regularly check for updates and security patches for all dependencies.