

# **FREE LABOUR**

Configuration/Release Documentation

7/4/2025  
Version 1.0

# TABLE OF CONTENTS

---

<b>TABLE OF CONTENTS.....</b>	<b>1</b>
<b>1. DOCUMENT INFORMATION.....</b>	<b>2</b>
1.1 Revision History.....	2
1.2 Approval.....	2
<b>2. FRONTEND INSTALLATION/SETUP.....</b>	<b>3</b>
2.1 Development Environment.....	3
2.2 Testing Environment.....	4
2.3 Production Environment.....	4
<b>3. BACKEND INSTALLATION/SETUP.....</b>	<b>4</b>
3.1 Backend Architecture.....	4
3.2 Development Environment.....	5
3.3 Environmental Variables.....	6
3.4 Azure Descriptions.....	6

# 1. DOCUMENT INFORMATION

---

## 1.1 Revision History

Date	Version	Status	Prepared by	Comments
4/3/2025	0.1	Template	Alexander Yang	
7/3/2025	1.0	Complete	Full Team	

## 1.2 Approval

Role	Name	Signature	Sign-off Date
Project Manager	Alexander Yang	A.Y.	7/4/2025
Full Stack Lead	Dhruv Khanna	D.K.	
Instructor	Jerry Jim		7/4/2025
Teaching Assistant	Shashank Hosahalli Shivamurthy		7/4/2025
Teaching Assistant	Mohsen Salehi		7/4/2025
Teaching Assistant	Jennie Chen		7/4/2025
Teaching Assistant	Malinda Gunathilaka		7/4/2025
Teaching Assistant	Mobina Shahbandeh Vayghan		7/4/2025
Backend	Yunze Guan	Y.G.	7/4/2025
Full Stack Programmer	Mukund Patil	M.P.	7/4/2025
Frontend Lead	Jasmine Mann	J.M.	7/4/2025
Frontend Programmer	Mandy Deng	M.D.	7/4/2025

Backend Programmer	Pranjali Lal Das	P.D.	7/4/2025
Backend Lead	Helmi Rouf	H.R.	7/4/2025
Sponsor	Steve Robinson		7/4/2025
Sponsor	Abhi Baro		7/4/2025
Sponsor	Nash Naidoo		7/4/2025

## 2. FRONTEND INSTALLATION/SETUP

---

### 2.1 Development Environment

The repository can be found at <https://github.com/CPSC319-2025/Team-6-AE>. Steps to launch the application are as follows:

1. Clone the repository onto a local machine
2. Inside a terminal with script access, navigate to `./frontend`.
3. Run the command `npm install` in the terminal to install dependencies.
4. Run the command `npm start` to launch the application. It will open in a browser window.
5. Test credentials are as follows:

Permissions	Name	Password
Admin	admin@gmail.com	password
User	user@gmail.com	password

6. The source code to be edited for development is located in `./frontend/src`.

## 2.2 Testing Environment

A test file containing automated scripts can be found at `./frontend/tests`. Steps to run it are as follows:

1. Inside a terminal with script access, navigate to `./frontend`.
2. Run the command `npx mocha tests/UserTests.spec.js`.
3. The test script will use the WIP link of <https://thankful-field-0410c1a1e6.azurestaticapps.net/#/login>.
4. Steps to update the link are below. Note that this should be changed for future use, as this is a temporary link not to be used long-term by sponsors.
5. Run the command `npm run build`. This will create a folder ready for deployment at `./frontend/build`.
6. On a machine with Azure Static Web Apps installed, run the command `swa deploy --env production`.

## 2.3 Production Environment

Steps to create a build ready for deployment are as follows:

1. Run the command `npm run build`. This will create a folder ready for deployment at `./frontend/build`.

# 3. BACKEND INSTALLATION/SETUP

---

## 3.1 Backend Architecture

1. Engine Layer: Handles core logic such as data processing, metadata extraction, image management

2. Controller Layer: Manages API endpoints using .NET 8, directing requests and responses between the engine and the client.
3. Model Layer: Defines structured data models used across the system, aligned with Azure SQL schema and API responses.

## 3.2 Development Environment

Prerequisites:

- .NET 8 SDK
    - Used as the primary backend framework to build RESTful APIs. Enables clean separation of concerns through a layered architecture.
  - AzureCLI
    - Azure Blob Storage
      - Stores uploaded images and videos. Integrated with metadata capture and file management workflows.
    - Azure SQL Database
      - Stores all structured data including metadata, tags, logs, user roles, and image references.
    - Azure App Services
      - Hosts the deployed backend (built with .NET 8) for scalable and managed access.
    - Azure API Management Services
      - Exposes .NET API endpoints securely for integration with external systems or frontend access.
  - Swagger
    - Integrated for API documentation and testing. Offers a web interface to interact with the .NET 8 controllers and test endpoint functionality.
4. Inside a terminal with of cloned repository, navigate to `cd src/Team-6-AE-DAM-Backend`
  5. Restore dependencies with `dotnet restore`
  6. Run migrations to update the SQL database `dotnet ef database update --context SQLDbContext`
  7. Start the backend with `dotnet run`

### 3.3 Environmental Variables

Ensure the following environmental variables are set appropriately in appsettings.json

- StorageAccount: Azure Blob Storage connection string
- AzureSQL: Azure SQL DB connection string
- THUMBNAIL\_CONTAINER: Name of container for thumbnail images
- PALETTE\_CONTAINER: Name of container for original upload

### 3.4 Azure Descriptions

Blob Storage

- Stores original and thumbnail image files.
- Organized in separate containers

Azure SQL

- Stores structured data for:
  - Users, Projects, Files
  - Tags, Metadata, Relations
- Relationships and constraints are managed via EF Core fluent configuration in SQLDbContext.cs