

Configuration Manual

MSc Research Project
Msc Research in Cloud Computing

Shreya Acharya
Student ID: x22115129

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Shreya Acharya
Student ID:	x22115129
Programme:	Msc Research in Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	24/04/2025
Project Title:	Configuration Manual
Word Count:	777
Page Count:	3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shreya Acharya
Date:	26th May 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Shreya Acharya
x22115129

1 Section 1

This manual documents the configuration for the two-part project setup:

- **Part 1:** Hosted on an EC2 instance (Node.js + EJS app)
- **Part 2:** Serverless backend functions (AWS Lambda via `serverless.yml`)

1.1 Part 1: EC2-Hosted App (Node.js + EJS)

1.1.1 Tech Stack

- **Backend:** Node.js + Express
- **Frontend:** EJS (Embedded JavaScript Templates)
- **Database:** MongoDB Atlas (Cloud-hosted)

1.1.2 Setup Instructions

- 1. Clone & Install Dependencies** `git clone repo-url cd project-directory npm install`
- 2. Environment Variables** Create a `.env` file: Sample Content
`PORT=3000 — MONGODB-URI= mongodb-uri — JWT-SECRET=jwt-secret —`
- 3. Start Server (Dev)** `npm start`

1.1.3 MongoDB Setup (EC2 App)

1. Used MongoDB Atlas for cloud-hosted database.
2. Create a new project and cluster.
3. Add EC2 server's IP to the **Network Access** → **IP Whitelist**.
4. Create a database user and get the **connection URI** (e.g.): `mongodb+srv://username:password`
5. We will use this URI in your `.env` as `MONGODB_URI`

1.1.4 PM2 Setup (Process Manager)

- 1. Install PM2 globally** `npm install -g pm2`

```

"dependencies": {
  "bcryptjs": "^2.4.3",
  "connect-flash": "^0.1.1",
  "ejs": "^3.1.5",
  "express": "^4.17.1",
  "express-ejs-layouts": "^2.5.0",
  "express-session": "^1.17.1",
  "jobspace": "file:",
  "moment": "^2.29.1",
  "mongoose": "^5.10.7",
  "multer": "^1.4.2",
  "nodemailer": "^6.4.14",
  "passport": "^0.4.1",
  "passport-local": "^1.0.0",
  "pm2": "^6.0.5",
  "prompt": "^1.0.0"
},

```

Figure 1: Dependencies involved the project

```

module.exports = {
  apps: [
    {
      name: "my-app",
      script: "npm",
      args: "run start",
      watch: false,
      env: {
        NODE_ENV: "development",
      },
      env_production: {
        NODE_ENV: "production",
      },
    },
  ],
};

```

Figure 2: pme ecosystem.config.js

2. Start your app with PM2 `pm2 start index.js`

3. Save the process list `pm2 save`

4. Auto-start PM2 on reboot `pm2 start`

5. Common PM2 Commands `pm2 list`

1.2 Part 2: Serverless App (AWS Lambda)

1.2.1 Overview

This part uses the Serverless Framework to deploy lightweight backend functions.

1.2.2 File Structure

`serverless-app/` `serverless.yml` `handler.js`

```

"dependencies": {
  "bcryptjs": "^2.4.3",
  "connect-flash": "^0.1.1",
  "ejs": "^3.1.5",
  "express": "^4.17.1",
  "express-ejs-layouts": "^2.5.0",
  "express-session": "^1.17.1",
  "jobspace": "file:",
  "moment": "^2.29.1",
  "mongoose": "^5.10.7",
  "multer": "^1.4.2",
  "nodemailer": "^6.4.14",
  "passport": "^0.4.1",
  "passport-local": "^1.0.0",
  "pm2": "^6.0.5",
  "prompt": "^1.0.0"
},

```

Figure 3: Serverless.yml

1.2.3 Setup & Deployment

- Install Serverless CLI

```
npm install -g serverless
```

- AWS Credentials

```
aws configure
```

- Deploy Functions

```
cd serverless-app serverless deploy
```

1.2.4 MongoDB Setup (Serverless)

- MongoDB Atlas is recommended.
- Whitelist 0.0.0.0/0| in Atlas to allow access from AWS Lambda.
- Stored MongoDB URI in AWS Secrets Manager or use environment variables in serverless.yml.

Example:

```
provider: environment: MONGODB_URI: ${env:MONGODB_URI}
```

1.3 Link to the code

- **Project on Server:** <https://github.com/Shreya66-ach/Final-Project-On-Serverless.git>
- **Serverless Project:** <https://github.com/Shreya66-ach/Final-Project-On-Serverless.git>

1.4 Link to the Website

<http://13.48.136.155:4000/>