# Project Deployment Guide

## Overview

This guide provides step-by-step instructions for deploying your Node.js-based recipe-sharing application. The application includes user authentication, recipe management, and a user-friendly interface for viewing, adding, and managing recipes. The deployment process involves setting up a server, configuring a database, and hosting the application for public access.

## Prerequisites

Before deploying the application, ensure the following prerequisites are met:

- Node.js and npm installed on your server or development machine.

- MySQL server installed and running.

- Git installed for cloning the repository.

- A cloud hosting account (e.g., AWS, Heroku, or DigitalOcean).

## Step 1: Setup the Server

1. Choose a hosting provider (e.g., AWS EC2, DigitalOcean Droplets, or Heroku).

2. Launch a new server instance with the desired configuration.

3. Connect to the server via SSH and update the system packages using `sudo apt update && sudo apt upgrade`.

## Step 2: Install Dependencies

1. Install Node.js and npm:

- Use the command `sudo apt install nodejs npm`.

2. Install MySQL server:

- Use the command `sudo apt install mysql-server`.

3. Secure the MySQL installation using `sudo mysql\_secure\_installation`.

## Step 3: Configure the Database

1. Log in to MySQL using `mysql -u root -p`.

2. Create a new database for the application:

- `CREATE DATABASE recipe\_sharing;`

3. Create a user and grant privileges:

- `CREATE USER 'appuser'@'localhost' IDENTIFIED BY 'password';`

- `GRANT ALL PRIVILEGES ON recipe\_sharing.\* TO 'appuser'@'localhost';`

## Step 4: Deploy the Application

1. Clone the repository:

- `git clone <repository-url>`

2. Navigate to the project directory:

- `cd recipe-sharing`

3. Install project dependencies:

- `npm install`

4. Configure the database connection in `db.js`.

## Step 5: Start the Application

1. Start the server:

- `node app.js` or use a process manager like PM2 for production.

2. Access the application in a browser using the server's public IP and port.

## Step 6: Configure a Reverse Proxy (Optional)

1. Install Nginx using `sudo apt install nginx`.

2. Configure Nginx to forward requests to the Node.js application.

## Step 7: Monitor and Maintain

1. Regularly update dependencies using `npm update`.

2. Monitor server logs for errors and issues.