

## Installation Guide

### Table of Contents

Introduction .....	2
Overview .....	2
Purpose .....	2
Getting Started.....	2
Prerequisites .....	2
Installation.....	3
Installation on Windows .....	3
Installation on Linux .....	26
Download codebase.....	34
Backend Setup.....	34
Frontend Setup .....	39

# Introduction

## Overview

License Manager is a comprehensive software solution designed to streamline and automate the process of managing software licenses for organizations. This project aims to simplify license tracking, reduce compliance risks, and optimize software asset management.

## Purpose

The purpose of the License Manager application is to provide organizations with a centralized and automated solution for managing software licenses. By using this application, organizations can ensure compliance with licensing agreements, minimize the risk of non-compliance, and optimize their software asset management. The application aims to streamline license tracking, automate notifications for expiring licenses, and export license CSV, role-based authorization to keep it secure, and statistical representation of license data, all of which contribute to efficient and effective license management.

# Getting Started

## Prerequisites

To install and use the License Manager, please ensure the following prerequisites are met:

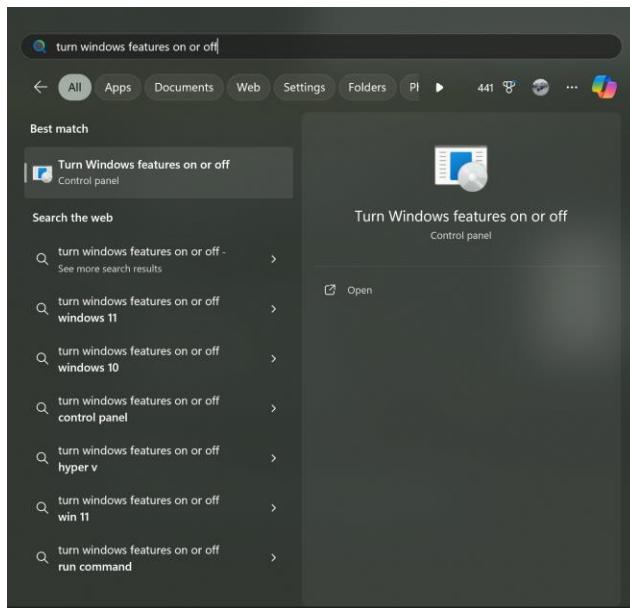
- **Operating System:** Windows 10 or later
- **Database:** MySQL version 8.0 or later
- **Runtime Environment:** Node.js version 20.11.1 or later, with npm version 10.2.4 or later
- **Web Server:** One of the following: Nginx, Apache, or IIS

## Installation

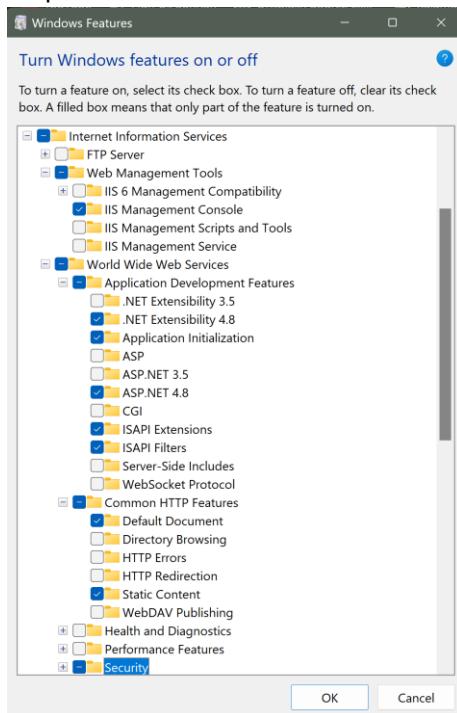
### Installation on Windows

#### Installation of web server

- IIS web server: To install IIS on windows 10, follow the steps below.
- Step 1: Search in start menu “Turn windows features on or off”



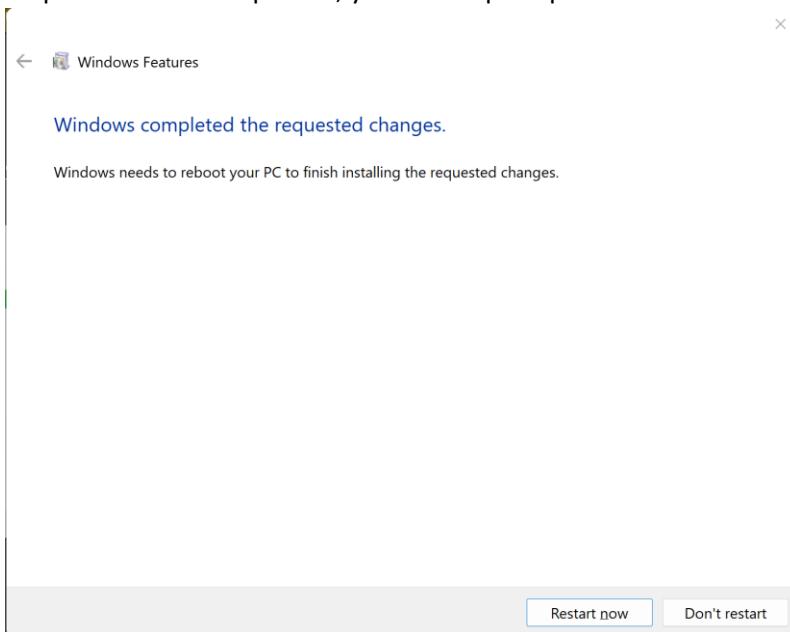
- Step 2: Find in the menu “Internet Information Services” and select the highlighted features.



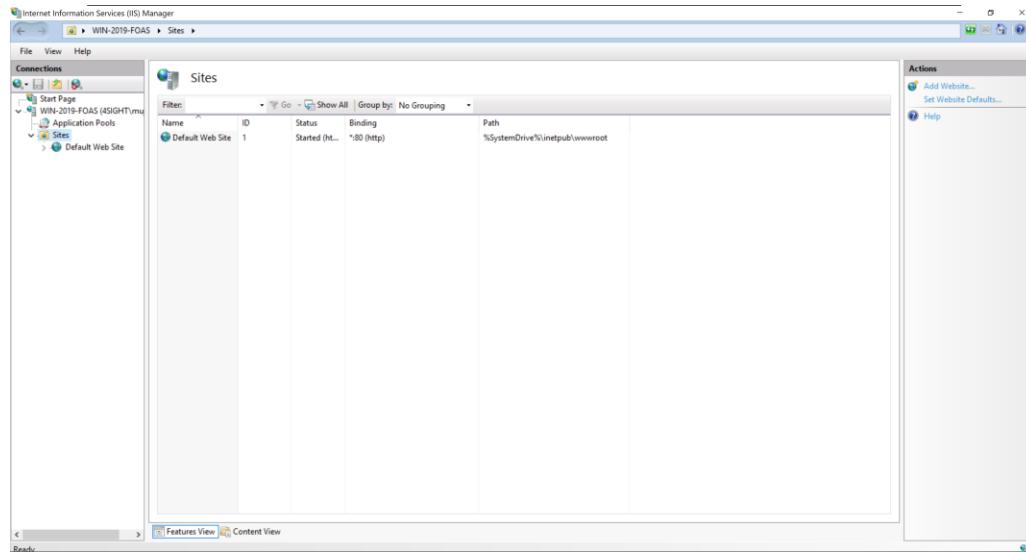
- Step 3: Press “OK”, you will see a modal saying “Applying Changes”



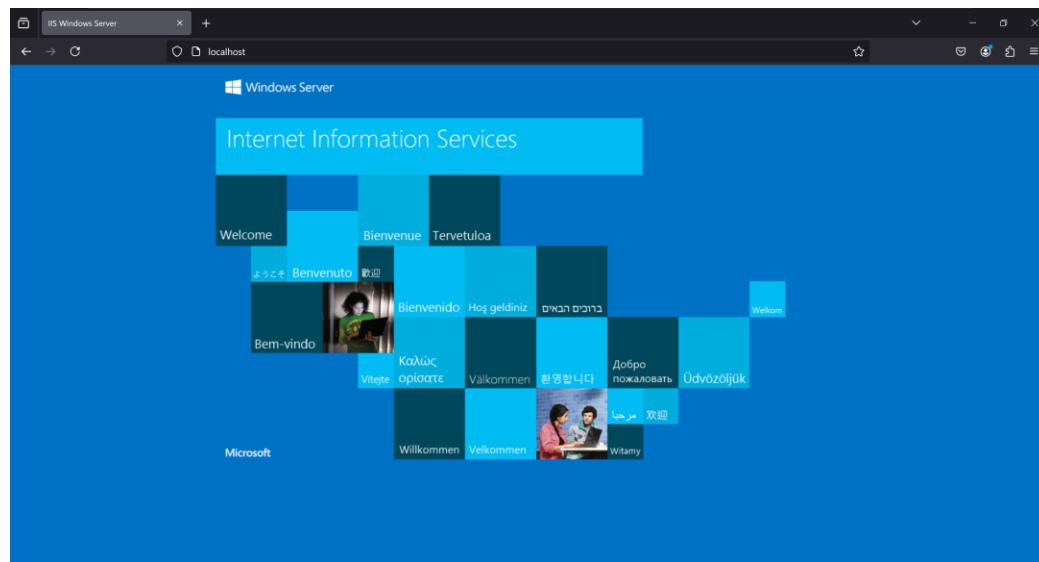
- Step 4: After it's completion, you will be prompted to reboot to finalize the changes.



- Step 5: After rebooting, you can open IIS by searching in the start menu. By default, there is default web site in IIS.



- Step 6: You can confirm the installation by opening your browser and writing in the address bar “localhost”. If you see the following screen, that means your web server IIS is setup successfully.



## Installation of MySQL and creating empty database

- Step 1: The simplest and recommended method is to download MySQL Installer for Windows from <https://dev.mysql.com/downloads/installer/> and execute it. Note that version 8.0 or later is recommended for the license manager.

The screenshot shows the MySQL Installer 8.0.23 download page. At the top, there are tabs for "General Availability (GA) Releases" (which is selected), "Archives", and a help icon. Below the tabs, the title "MySQL Installer 8.0.23" is displayed. A dropdown menu labeled "Select Operating System:" shows "Microsoft Windows" as the current selection. To the right of the dropdown is a link "Looking for previous GA versions?". The main content area lists two download options:

Version	File Type	Size	Action
8.0.23	Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.23.0.msi)	2.4M	<a href="#">Download</a>
8.0.23	Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.23.0.msi)	422.4M	<a href="#">Download</a>

Below the download links is a note: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download." It includes MD5 checksums and signature links for both files.

Note: On the next page, you'll be prompted to log in or create a MySQL account. You can choose "No thanks, just start my download."

[Login Now](#) or [Sign Up](#) for a free account.

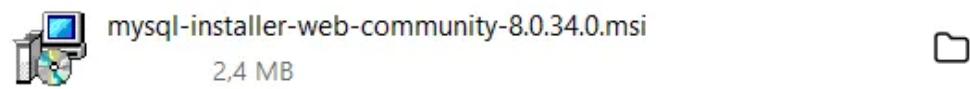
An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

The screenshot shows the MySQL login/signup page. It features two prominent buttons: a blue "Login »" button with the sub-instruction "using my Oracle Web account" below it, and a green "Sign Up »" button with the sub-instruction "for an Oracle Web account" below it. Below these buttons, a note explains the Oracle SSO authentication process: "MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions."

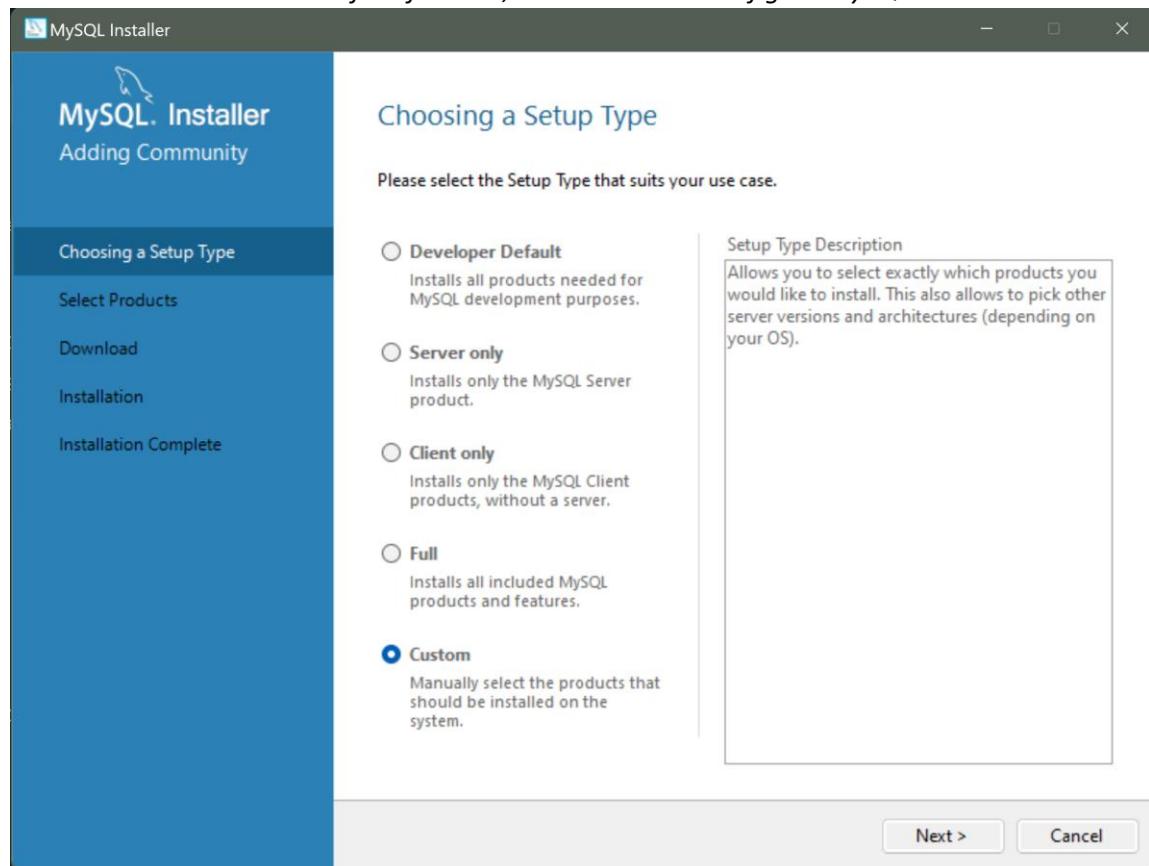
[No thanks, just start my download.](#)

- Step 2: Locate the installer file you just downloaded (e.g., mysql-installer-web-community-x.x.xx.x.msi) and double-click it to run the installer.

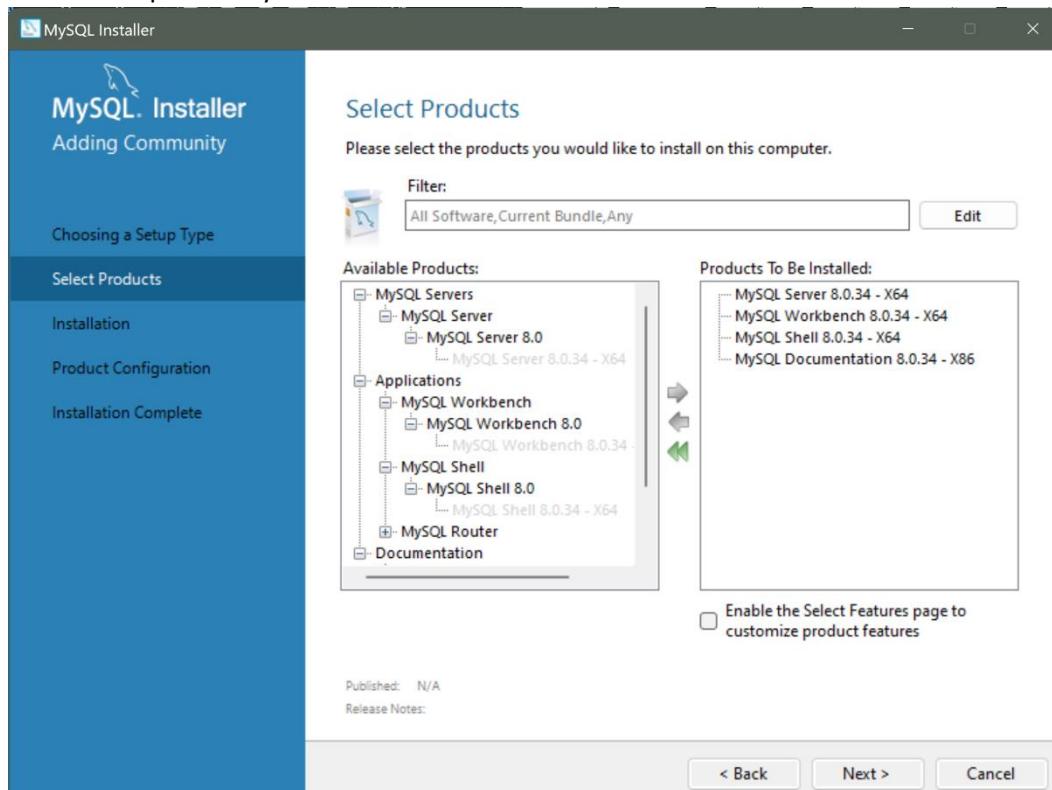


- Step 3: In the installer window, choose “Custom” under “Setup Type.” This option will let us choose the installation content MySQL Server, MySQL Workbench and other components, such as documentation and examples. Then click “Next.”

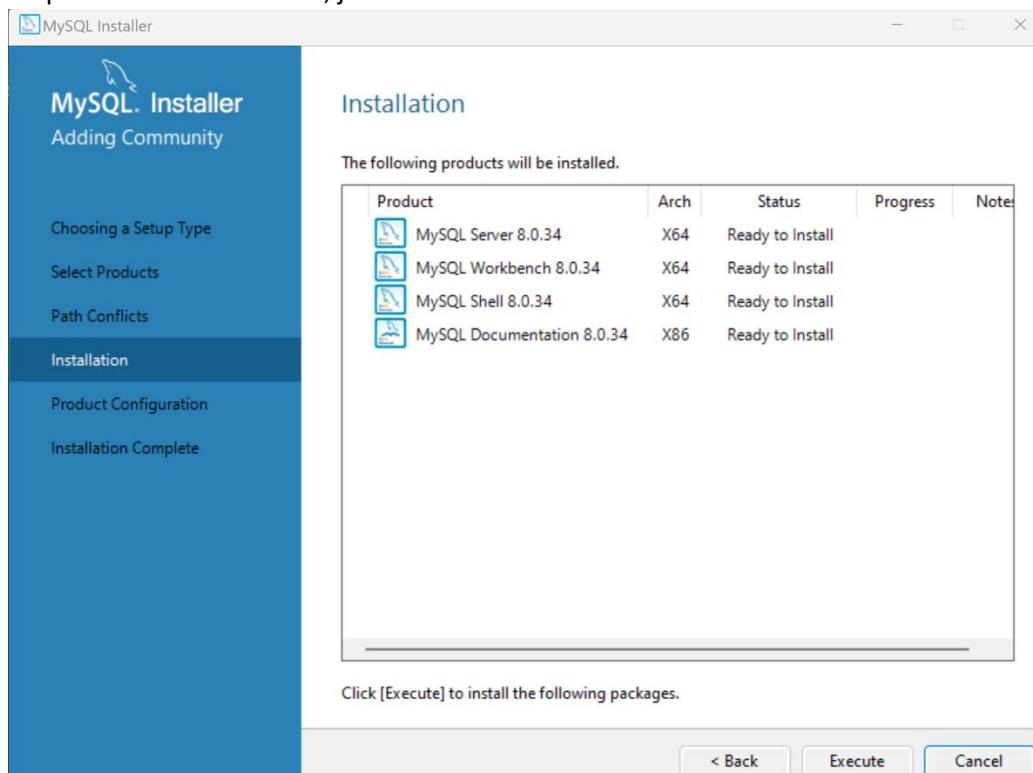
*\*Note: You can also choose “full” for ease, but will need to configure MySQL router as well.*



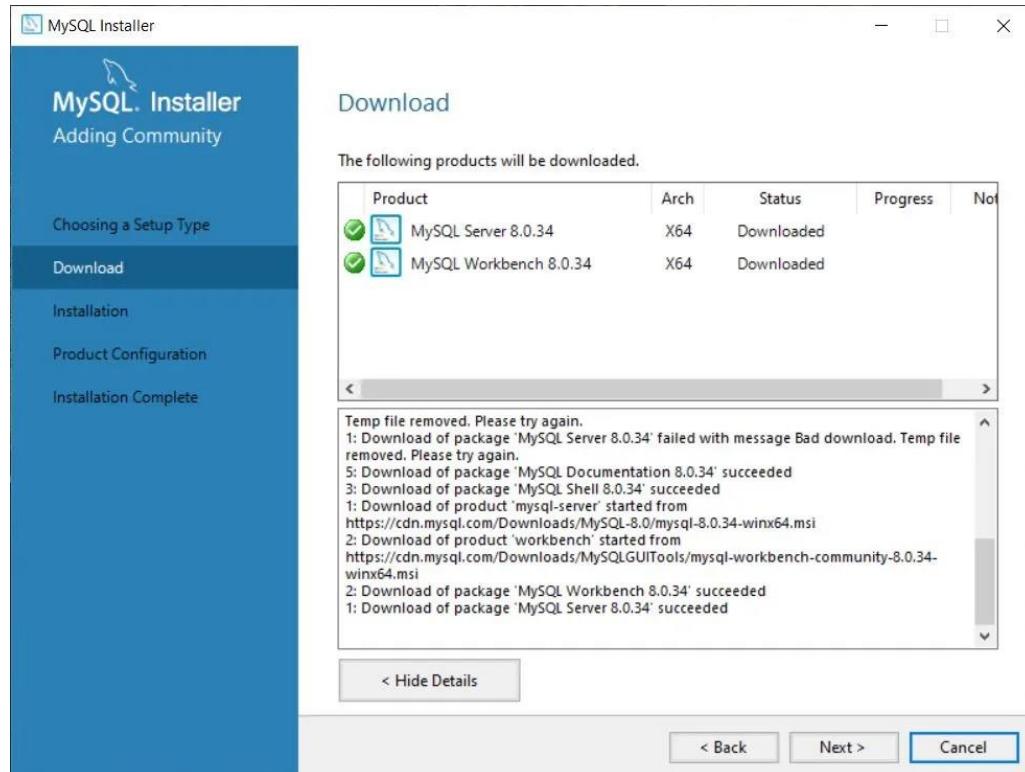
Choose the products you want to install as shown below and then click “Next”



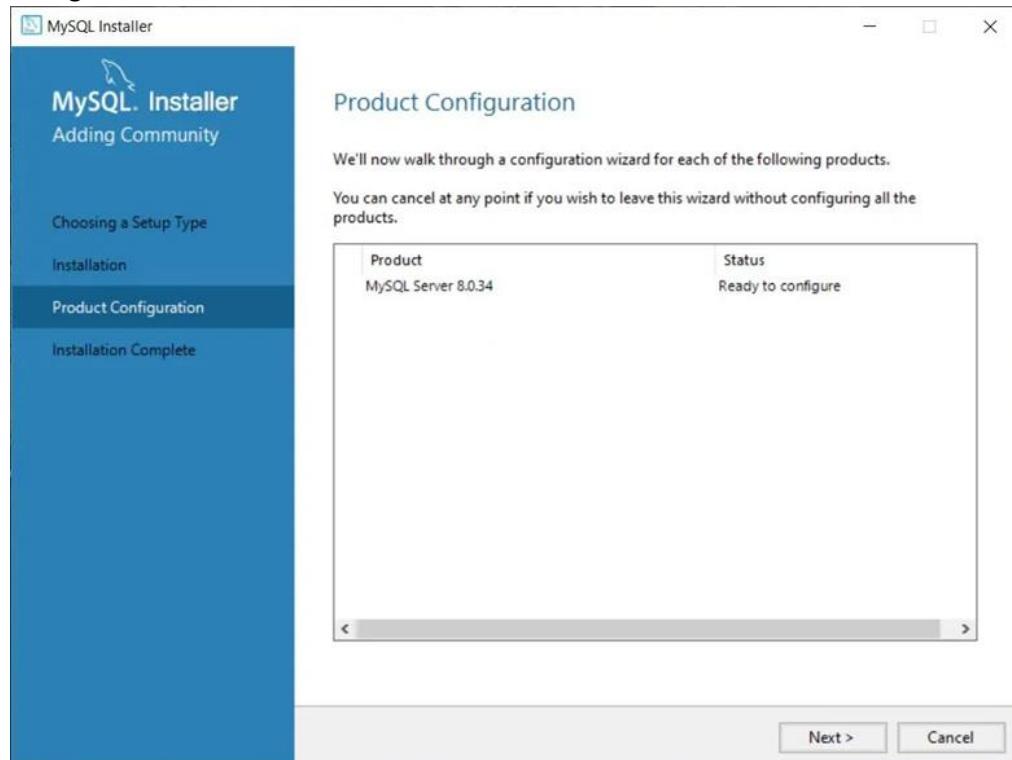
- Step 4: In the next window, just click “Execute.”



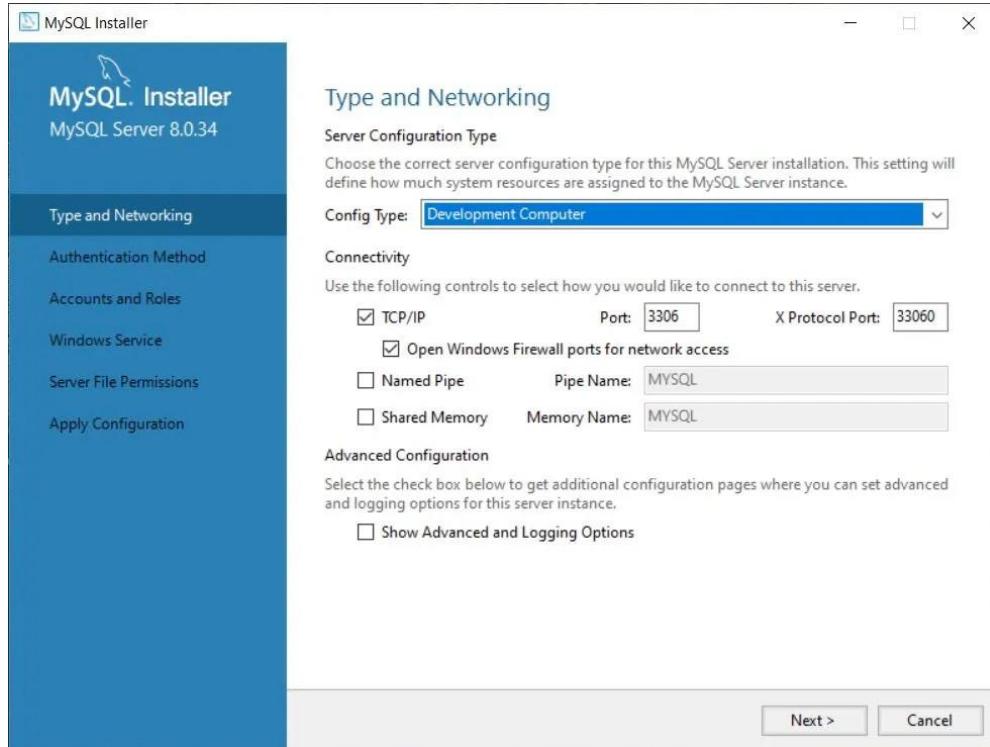
- Step 5: After all products have been downloaded sucessfully, as indicated by the green checkmark, click “Next.”



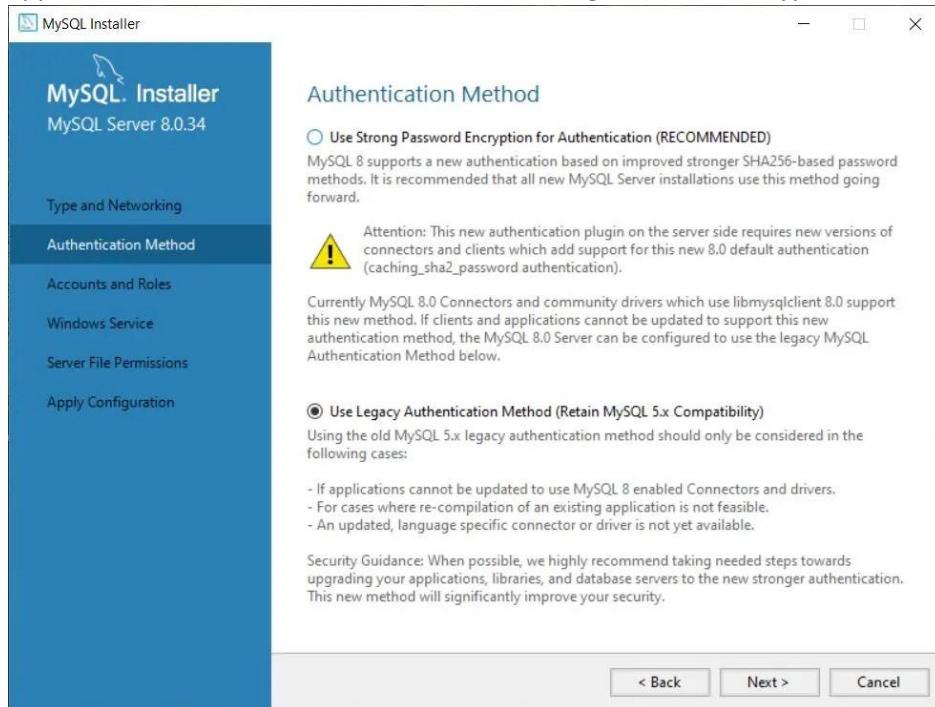
- Step 7: After all products have been installed successfully, click “Next” to start the configuration.



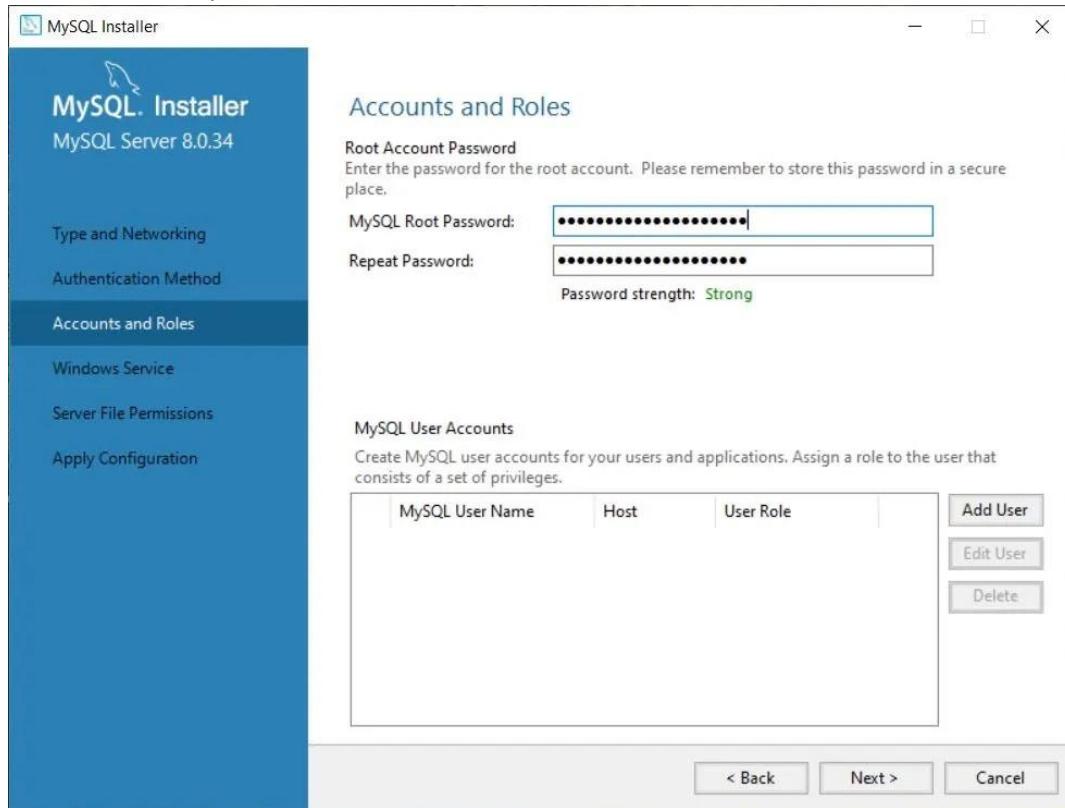
- Step 8: Under “Type and Networking” choose “Development Computer” as Config Type and leave everything else as is. Then click “Next.”



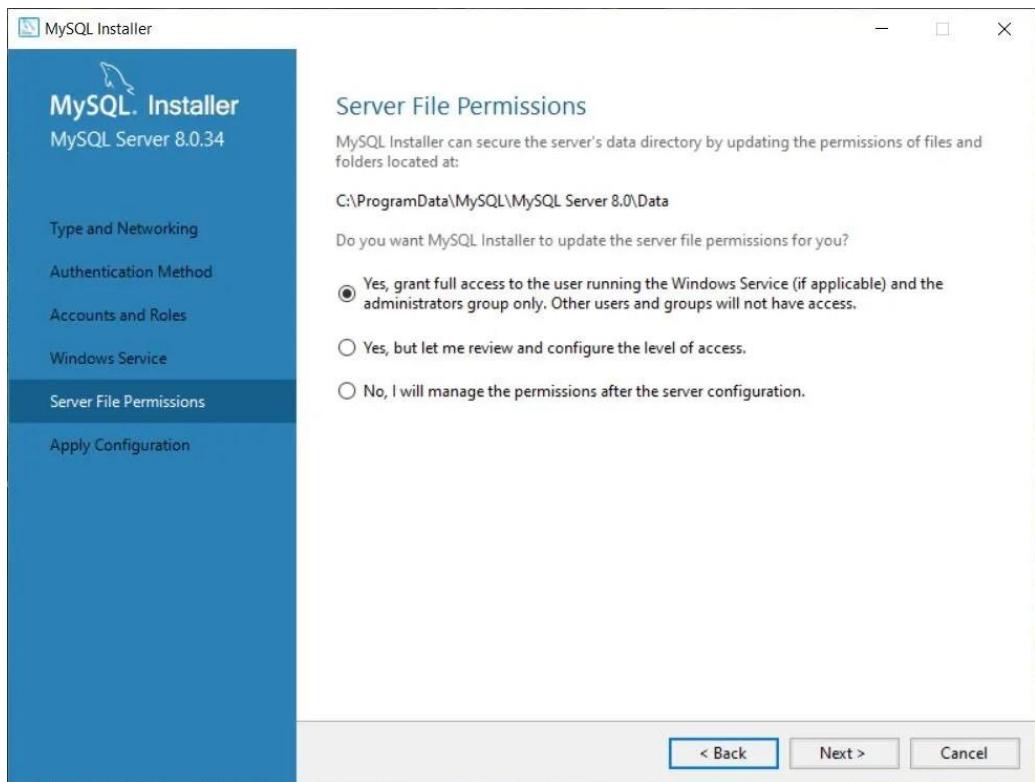
- Step 9: Next, you will be asked to choose an authentication method. For increased compatibility, choose “Use Legacy Authentication Method” and click “Next.” Choosing “Use Strong Password Encryption” will cause failure in Authentication through external applications. Therefore, **do not** select “Use Strong Password Encryption”



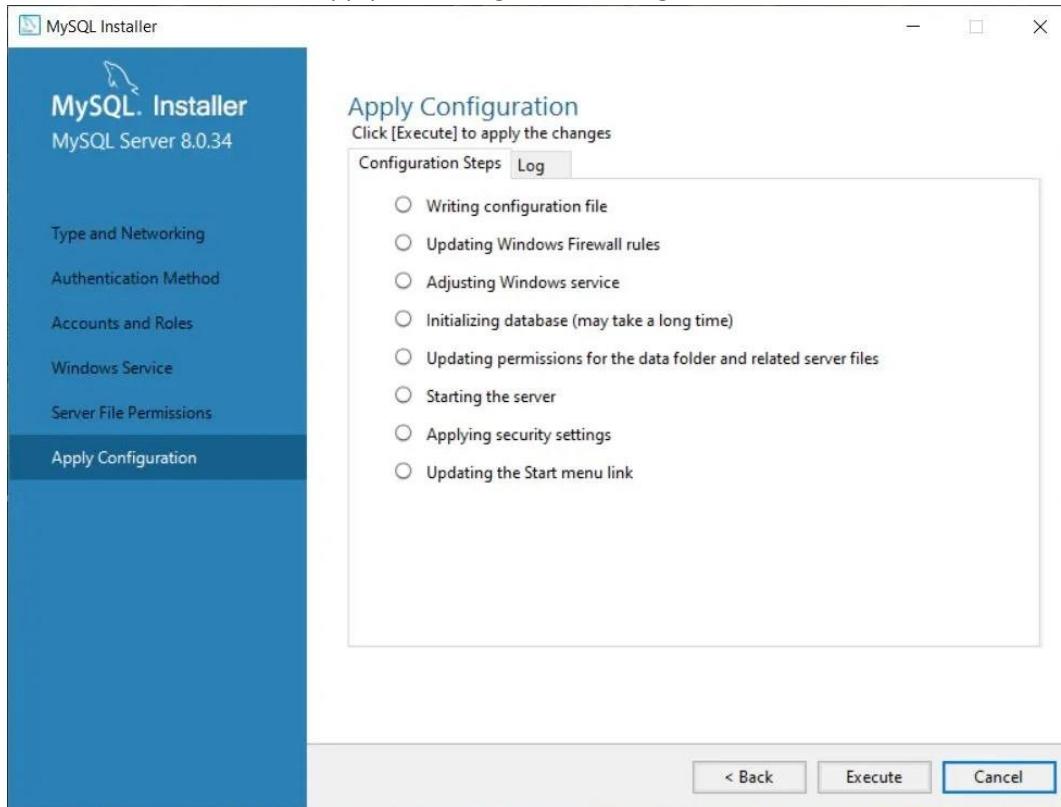
- Step 10: In the “Account and Roles” window, choose a new password for your root account. Be sure to choose a strong **password** and enter the password under MySQL Root Password and Repeat Password then click “Next.”



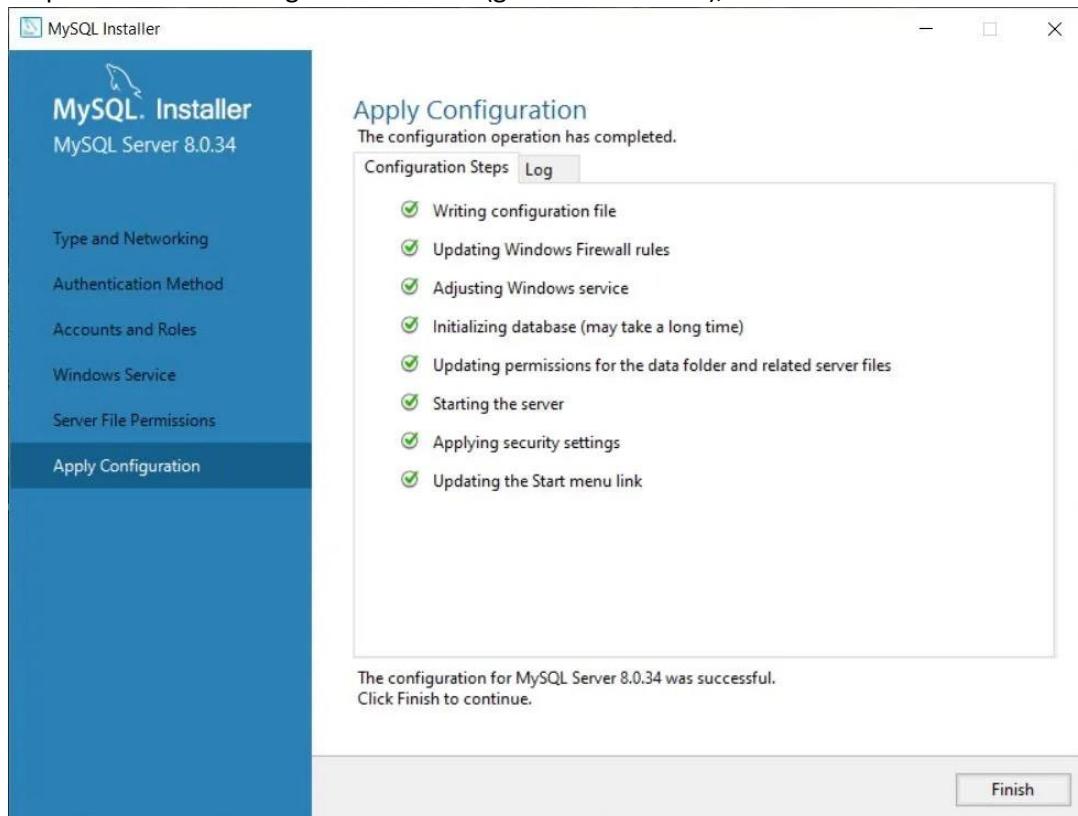
- Step 11: In the next window, leave everything as is and click “Next.”



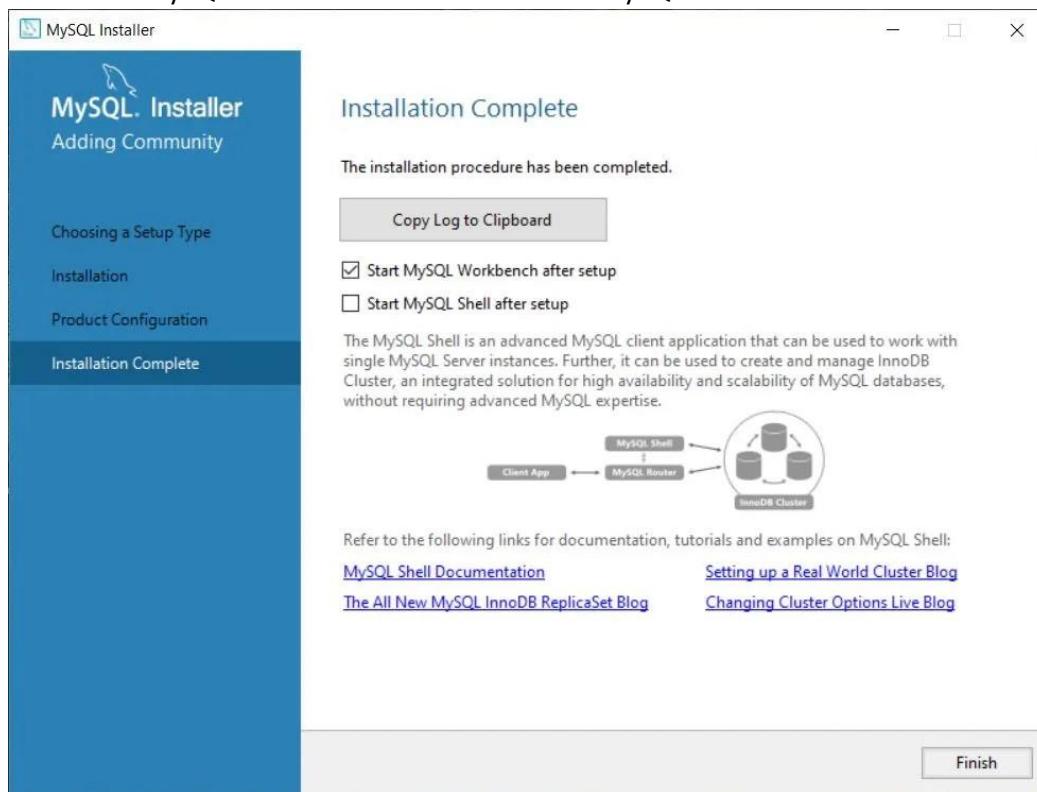
- Step 12: Next, grant full access to let the Installer update server file permissions and click "Next." Click "Execute" to apply the configuration changes.



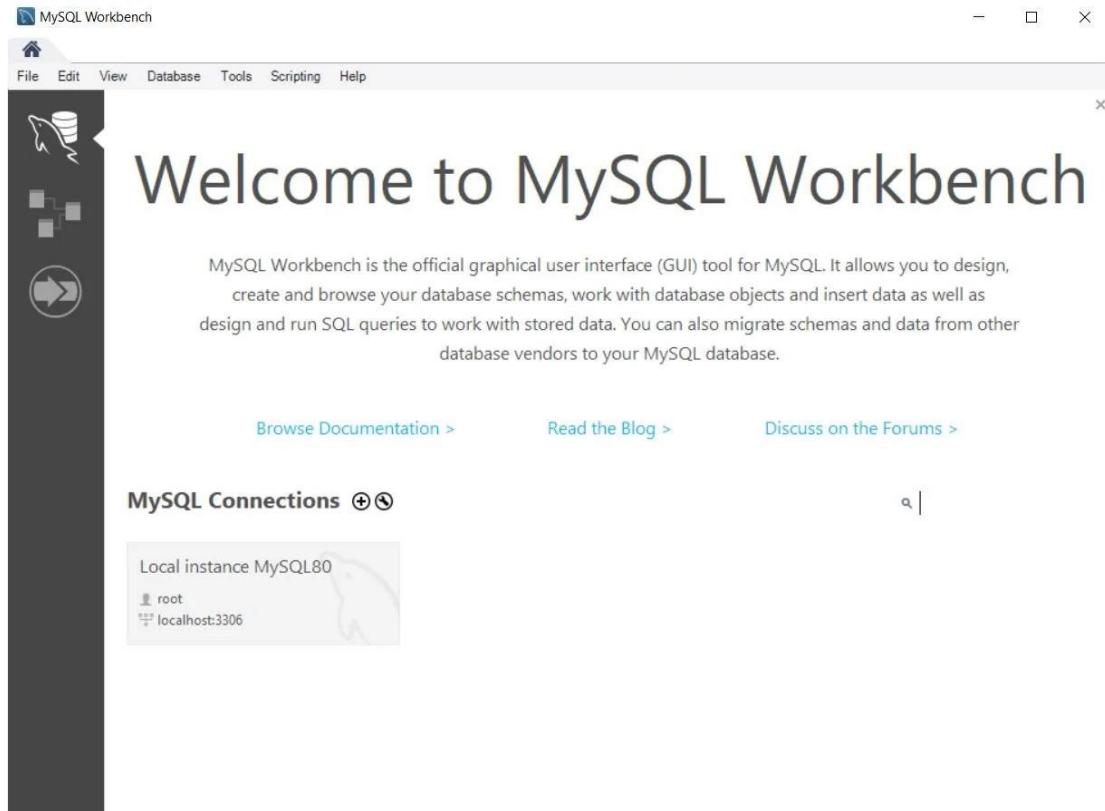
- Step 13: Once the configuration is done (green checkmarks), click “Finish.”



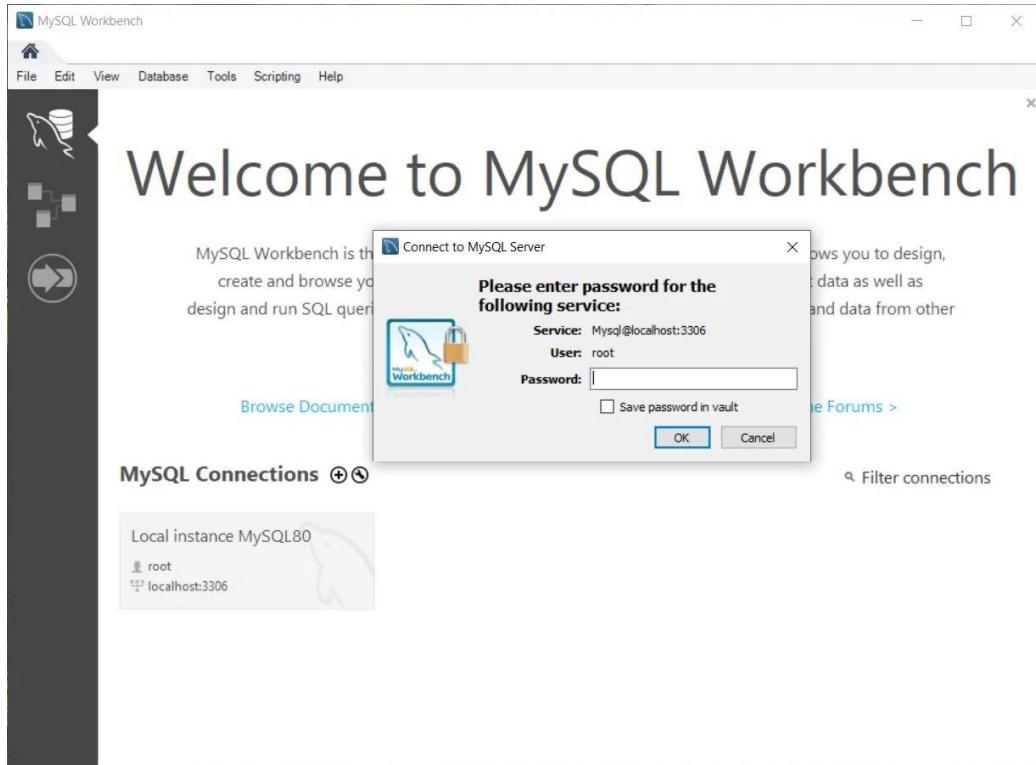
- Step 14: That's the installation and setup done. You just downloaded and installed MySQL Server and MySQL Workbench. Click “Finish” and MySQL Workbench will launch.



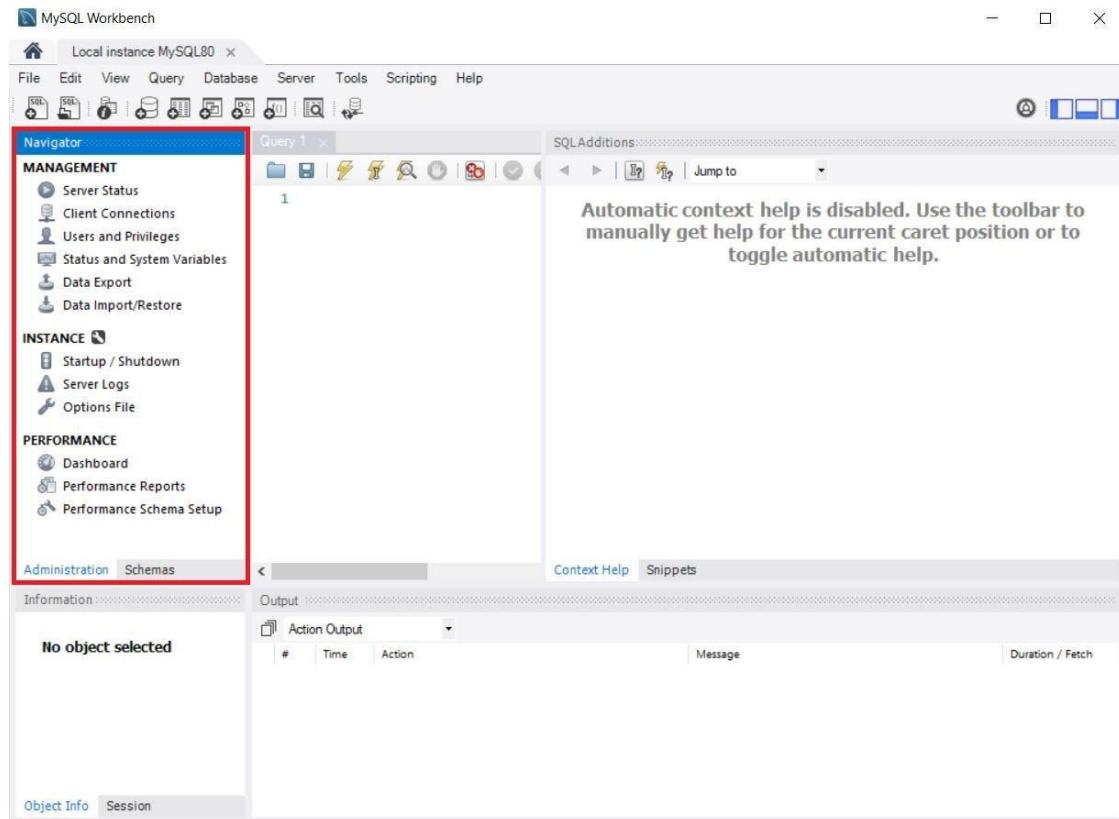
- Step 15: At the end of the installation process, MySQL Workbench should launch automatically. If it doesn't, start the program manually. You are greeted by the graphical user interface (GUI) tool for MySQL, the MySQL Workbench. In the bottom half of the window under "MySQL Connections" you should see one tile for your local instance connection, Local instance MySQL80. Go ahead and double-click it.



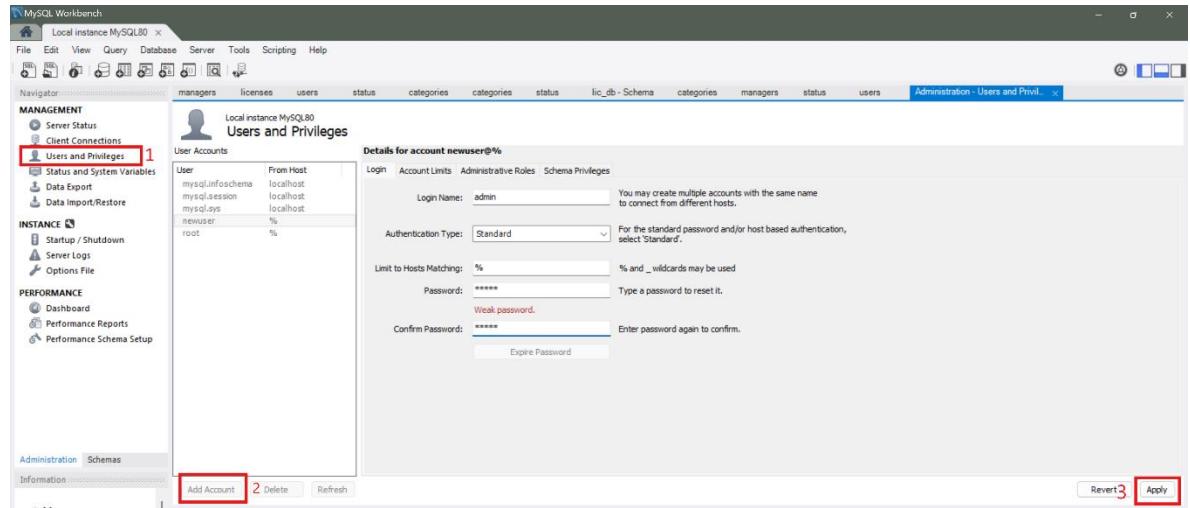
- Step 16: In the new window, enter the password you chose earlier and click "OK."



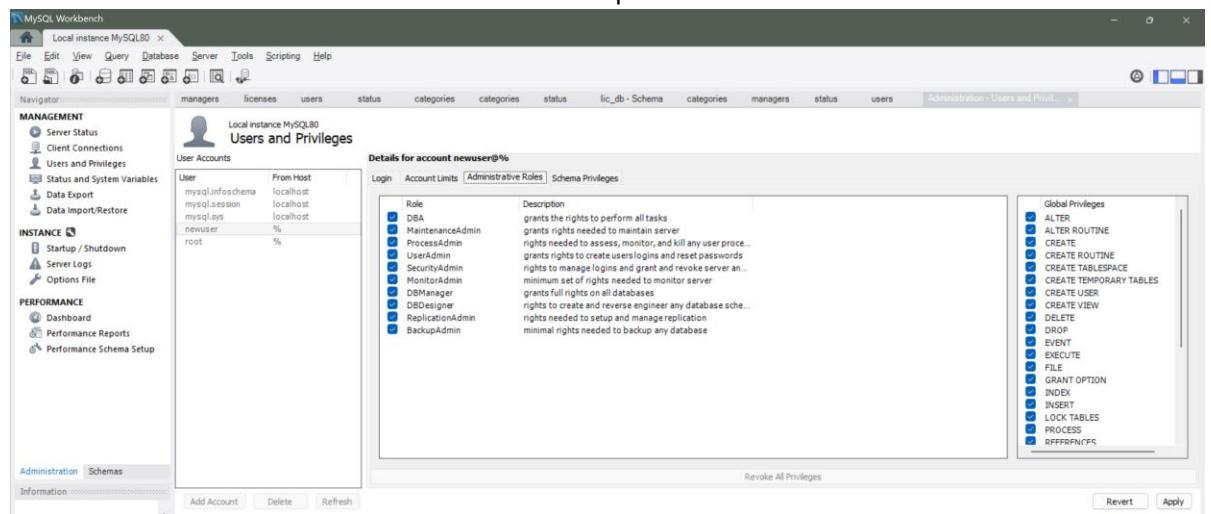
- Step 17: We're in. You are now connected to your MySQL Server and can start using the MySQL Workbench. The Navigator is typically located on the left-hand side of the MySQL Workbench GUI. It serves as a central hub for database management and navigation. It houses some administrative features and a handy tree-like structure of all your databases under the tab "Schemas."



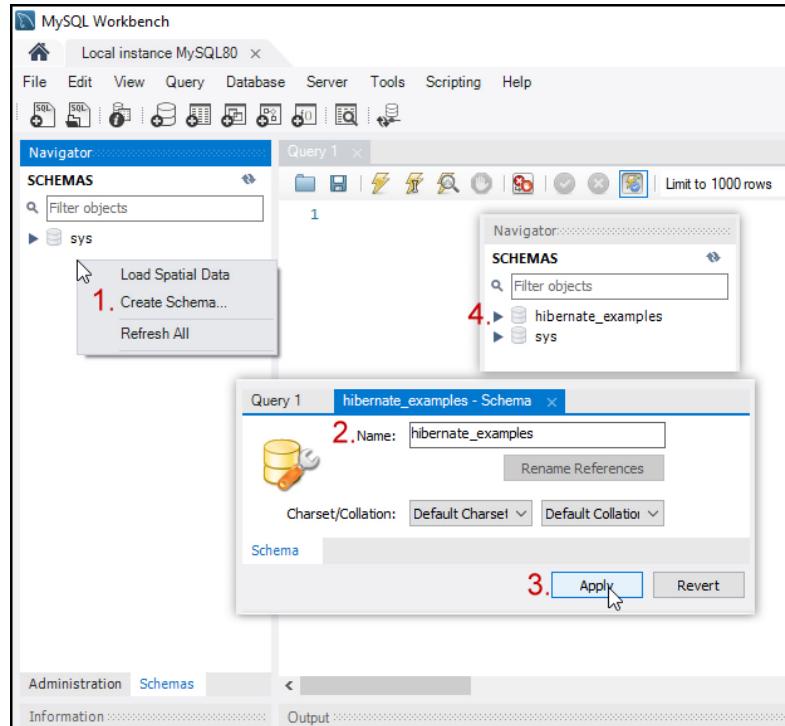
- Step 18: To enable remote connection from our application we need to create a new user with the host set to "%".



Further we need to set the user roles to ensure no permissions error occurs.



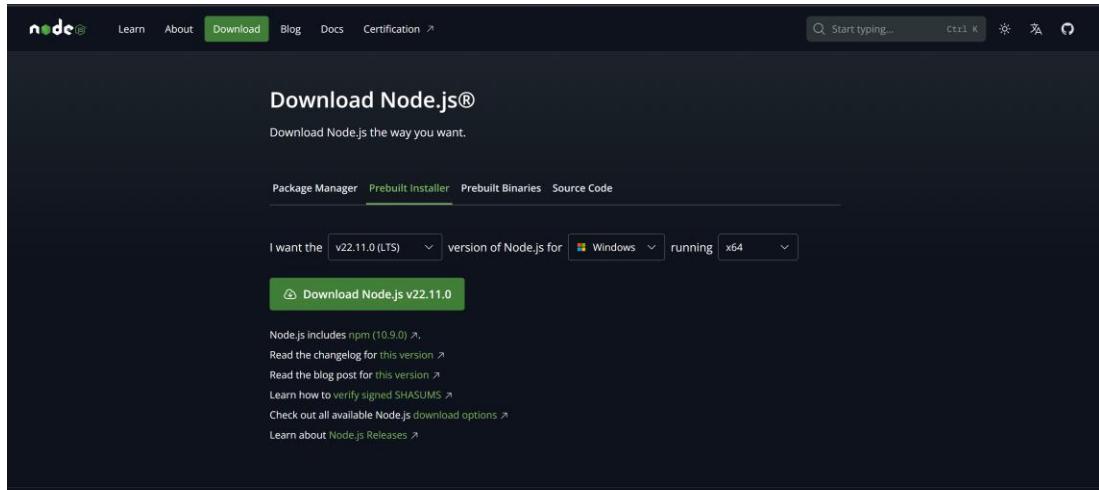
- Step 19: Navigate to the “Schemas” tab to see all your databases. To create a new database, click on the create new schema option in the top panel or right click in the empty space of the Schemas tab. Remember the database name, as it will be needed when configuring the application.



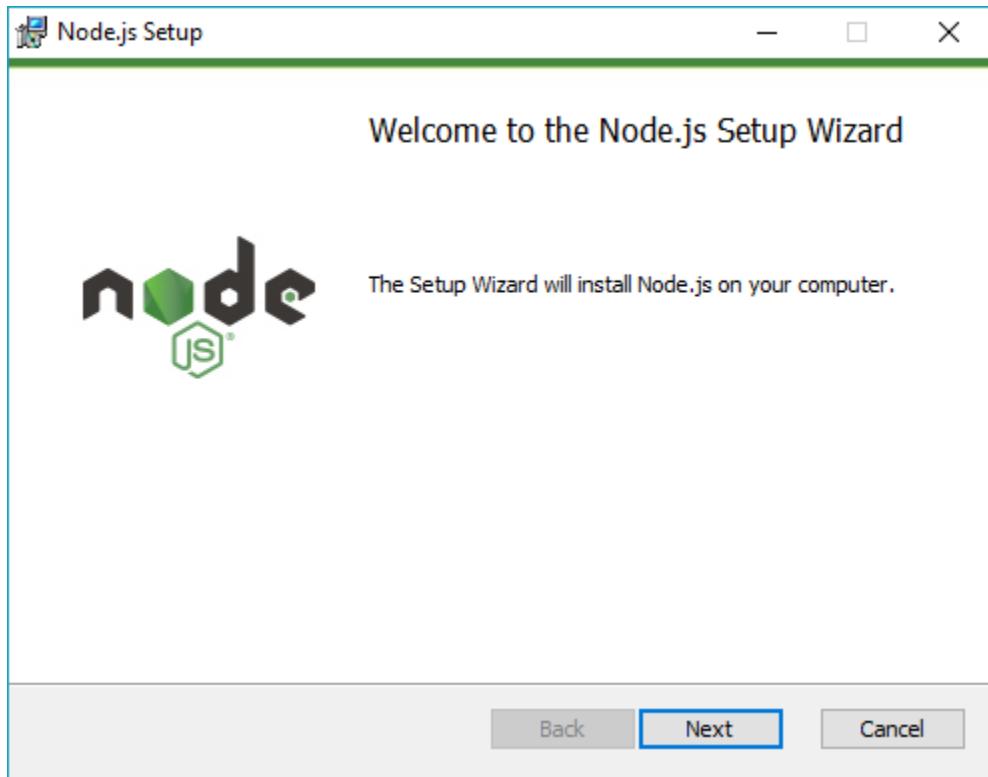
With this the installation of MySQL with its tools are installed and configured. Additionally, we have also initialized a database which we will need later.

### *Installation of node.js*

- Step 1: **Download the NodeJS.** Downloading the Node.js '.msi' installer the first step to install Node.js on Windows is to download the installer. Visit the official Node.js website i.e) <https://nodejs.org/en/download/>



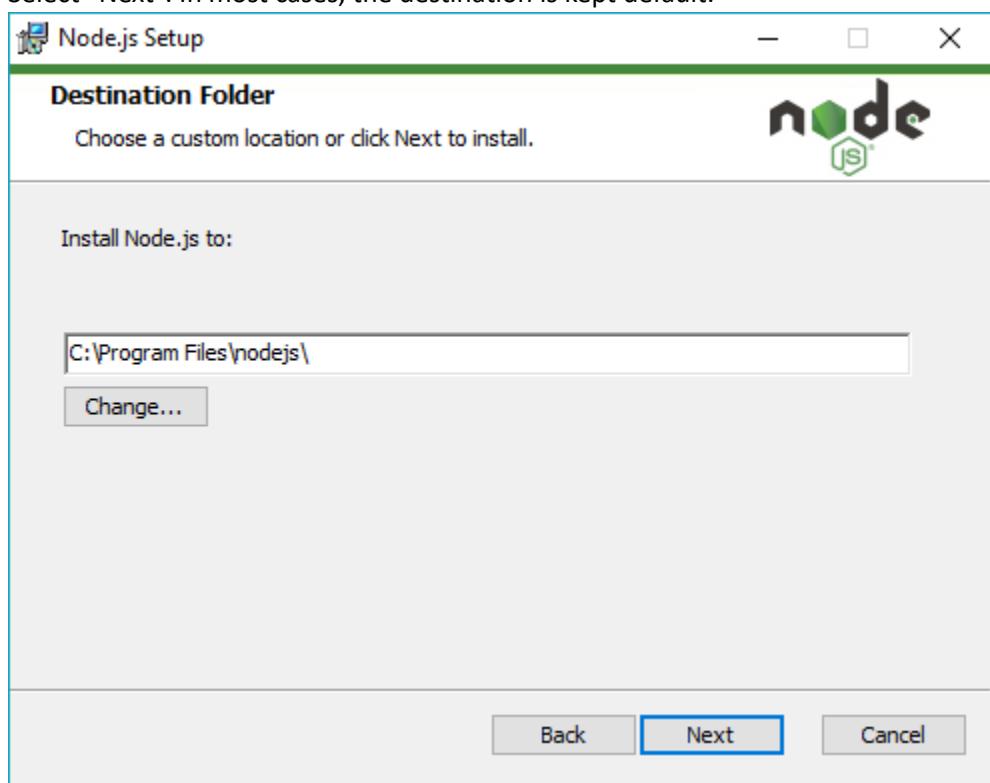
- Step 2: **Running the Node.js Installer.** Now you need to install the node.js installer on your PC. You need to follow the following steps for the Node.js to be installed: Double-click on the .msi installer



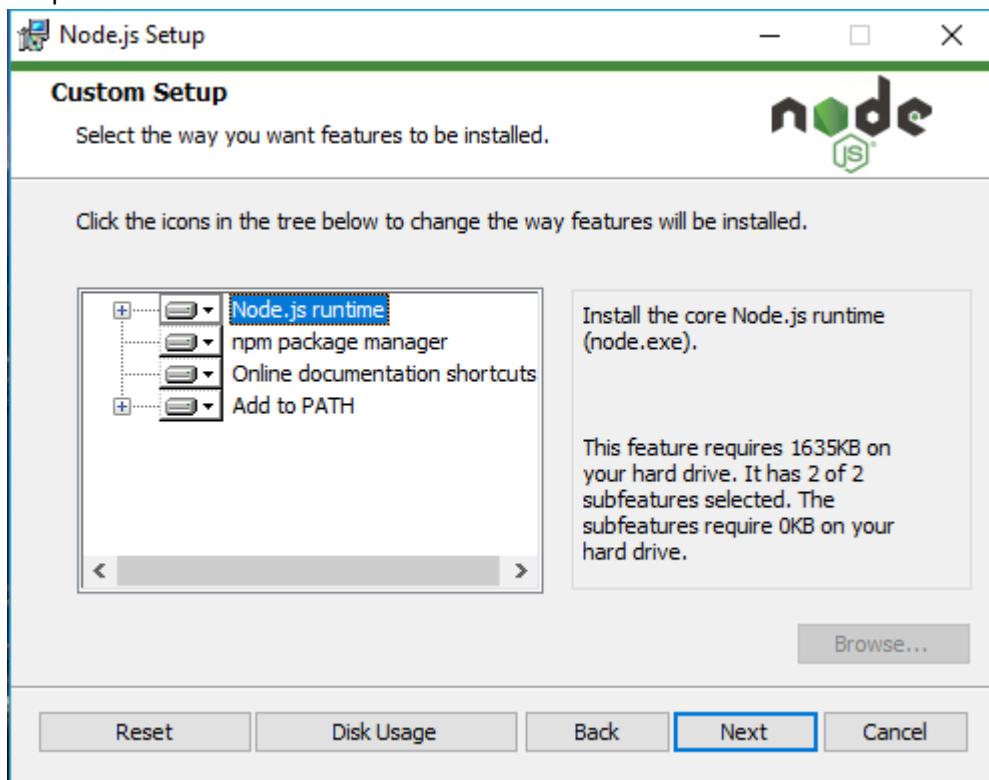
- Step 3: After clicking “Next”, the End-User License Agreement (EULA) will open. Check “I accept the terms in the License Agreement” and Select “Next”



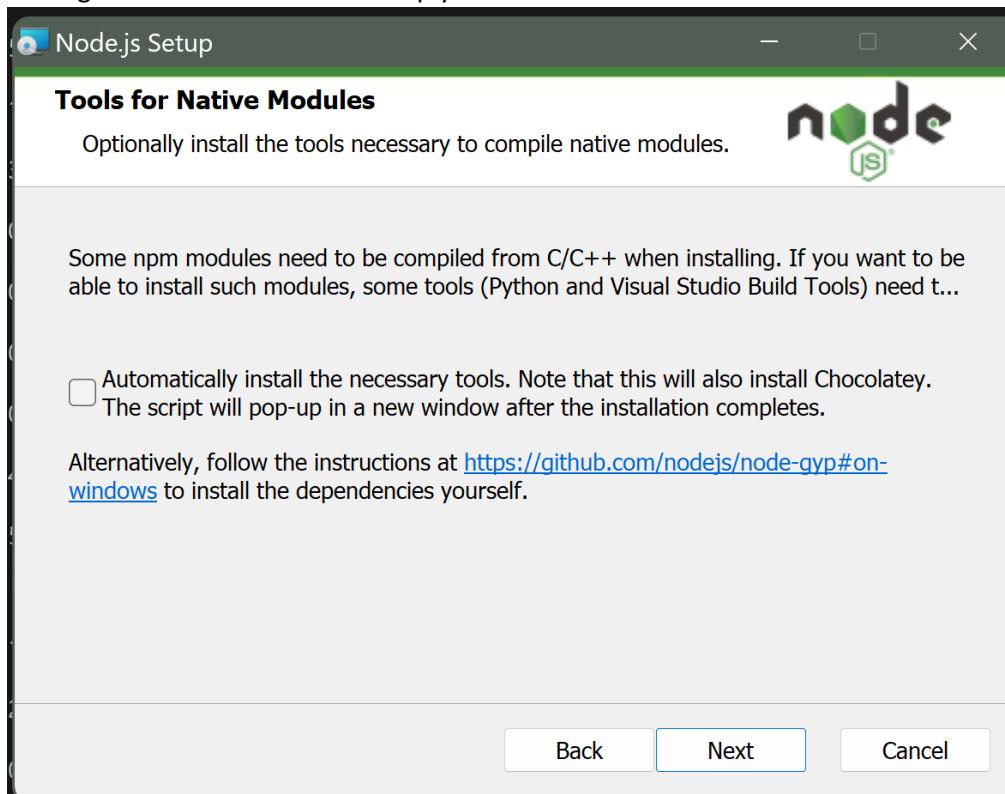
- Step 4: Destination Folder Set the Destination Folder where you want to install Node.js & Select “Next”. In most cases, the destination is kept default.



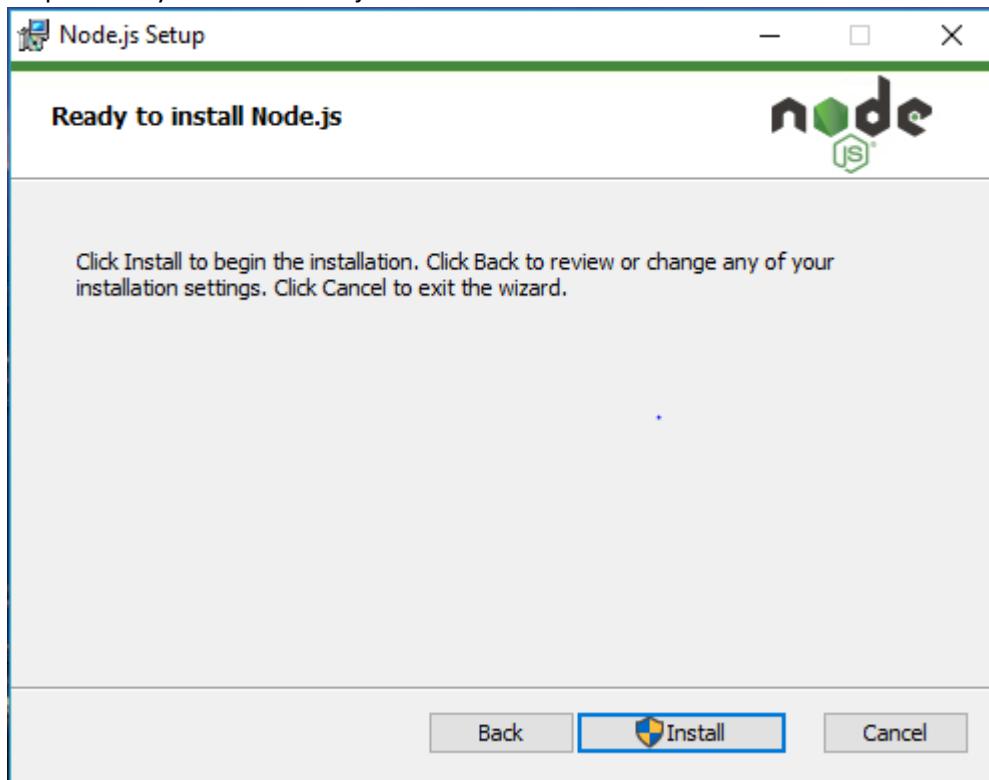
- Step 5: Select "Next"



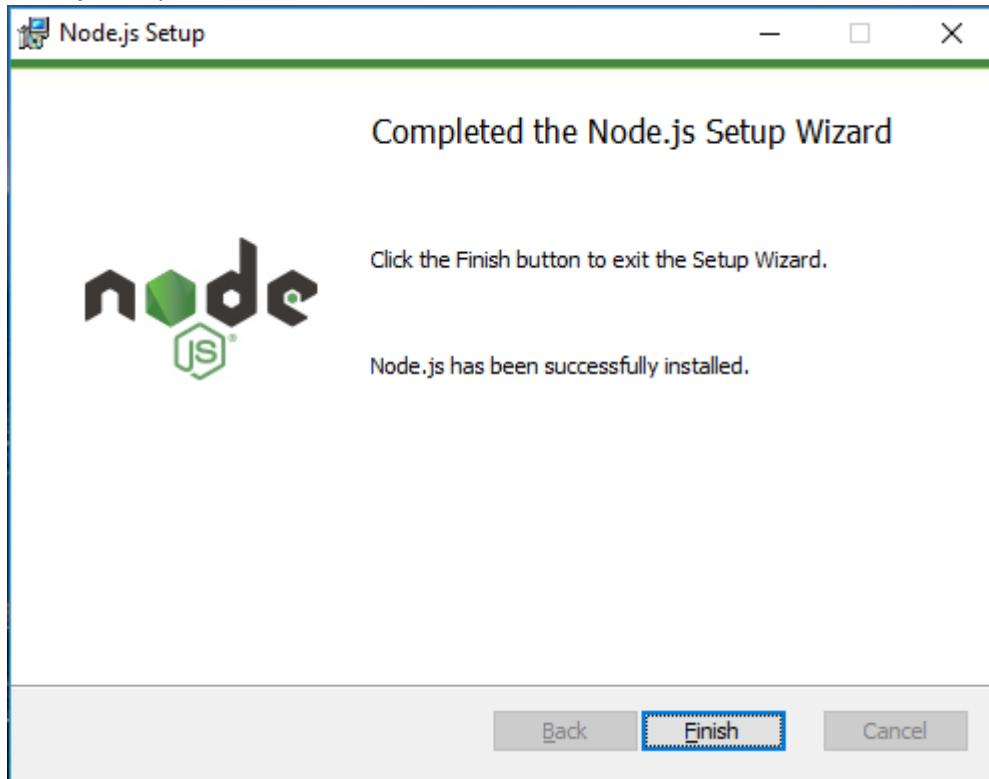
- Step 6: There is no need to install chocolatey and all necessary tools as that makes the setup use high amounts of resource. Simply Click "Next".



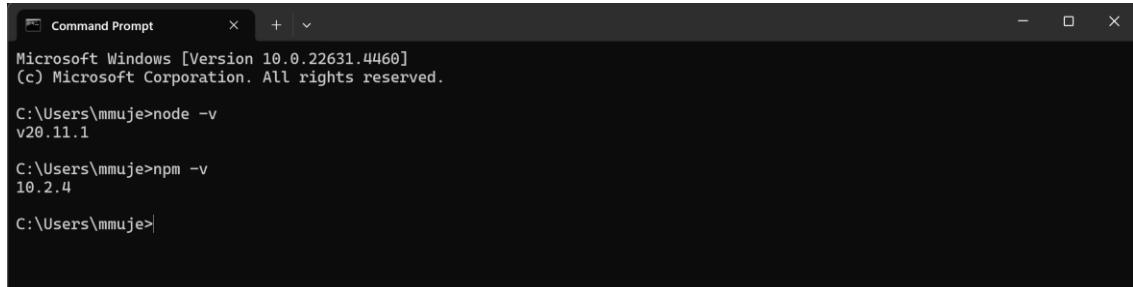
- Step 7: Ready to Install Node.js. Select “Install”



- Step 8: Do not close or cancel the installer until the installation is complete. Complete the Node.js Setup Wizard. Click “Finish”



- Step 9: Verify that Node.js was properly installed or not. To check that node.js was completely installed on your system or not, you can run the following command in your command prompt or Windows PowerShell and test it.



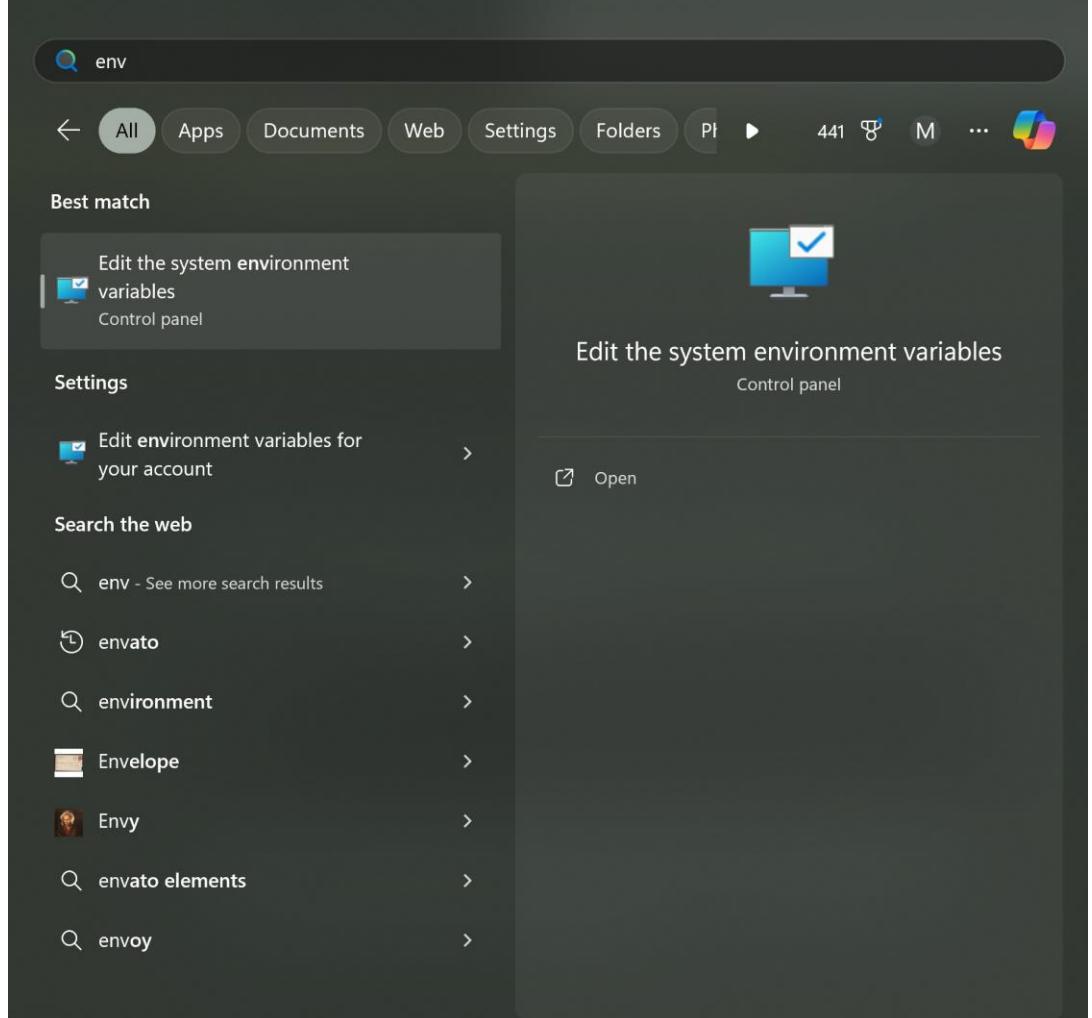
```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mmuje>node -v
v20.11.1

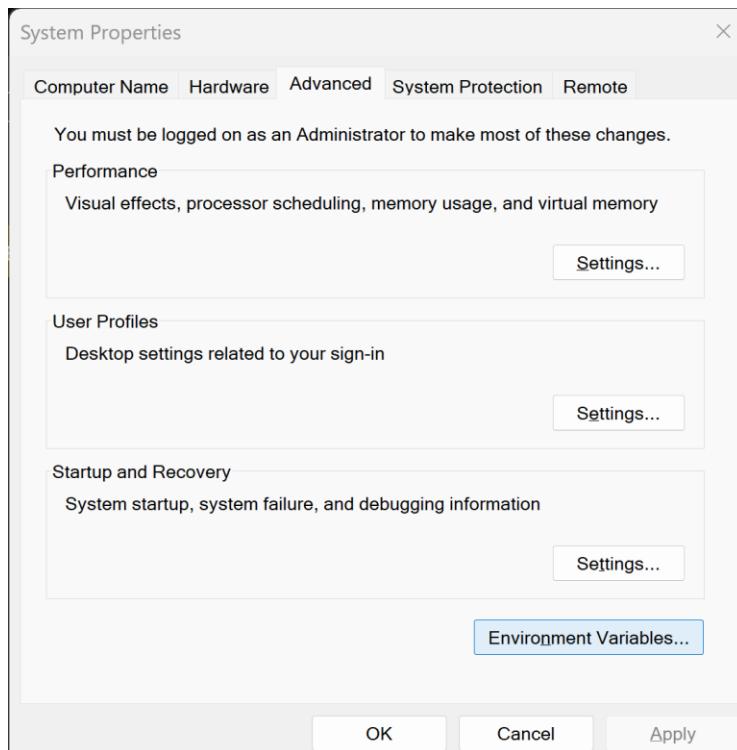
C:\Users\mmuje>npm -v
10.2.4

C:\Users\mmuje>
```

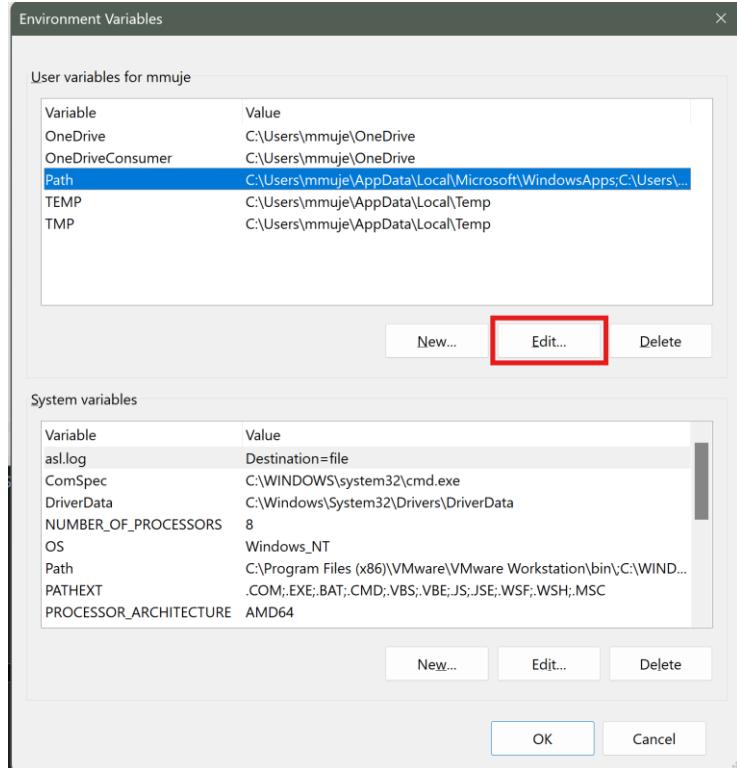
- Step 10: If node.js was completely installed on your system, the command prompt will print the version of the Node JS installed. If for some reason it gives you error regarding “node not recognized” either your installation was not successful, or if the installation was successful you only need to add the path to your node.js installation in the environment variables.
  - o Search in start menu “environment variables”



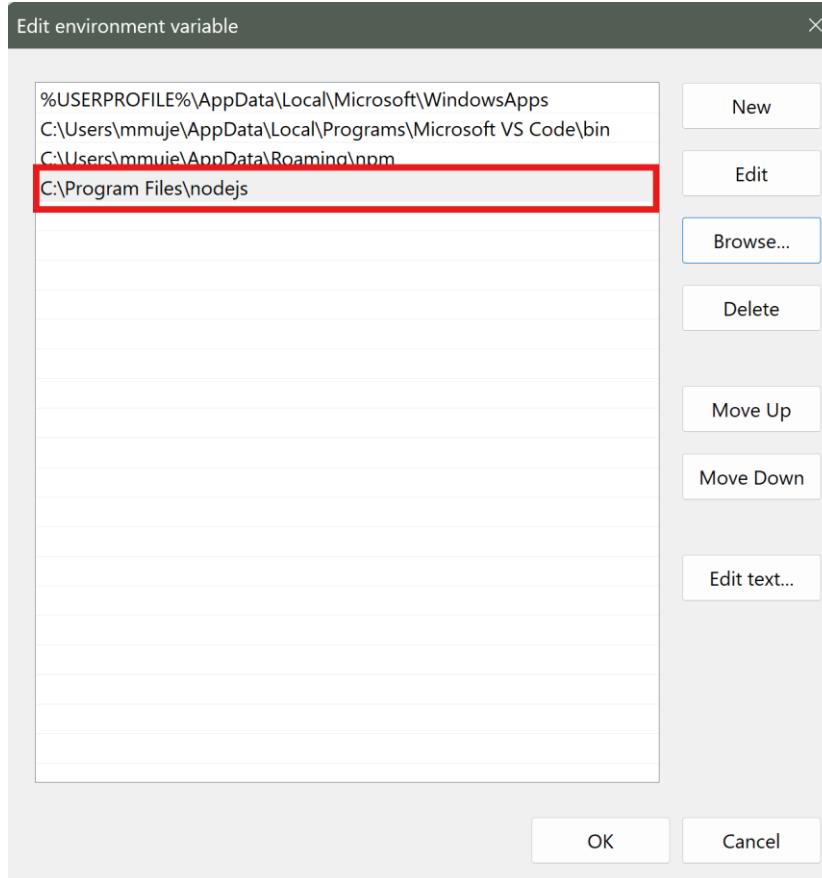
- Click on “Environment Variables”



- Select “Path” and click on “Edit”



- Click on “New” and then click on “Browse” to select the destination of where you installed your node.js.



- Click “OK” and close all the modals, this will save your settings. Now you can retry the Step 9 in node.js installation.

With this the installation of node.js is completed.

## Installation on Linux

Note: Keep in mind that installation might defer based on the linux distro on deployment environment. Additionally, before any package installation, do remember to update your system packages. If you are doing the installation at once, you only need to do it once.

### Installation of web server

- Nginx web server: Installing Nginx on linux machine, follow the steps below.
- Step 1: First step is to update your system packages.  
~~~~~

Ubuntu/Debian: sudo apt update

Centos/RHEL: sudo dnf upgrade

```
mujeeb@Mujeeb:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
mujeeb@Mujeeb:~$ |
```

- Step 2: Install nginx web server  
~~~~~

Ubuntu/Debian: sudo apt install nginx

Centos/RHEL: sudo dnf install nginx

```
mujeeb@Mujeeb:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 552 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1 [31.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.1 [521 kB]
Fetched 552 kB in 14s (38.5 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 40787 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7.1_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.1) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7.1_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx-common (1.24.0-2ubuntu7.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Processing triggers for man-db (2.12.0-4build2) ...
```

- Step 3: Enable and Start the nginx web server. Along with it we can check the status after the two command of our web server  
~~~~~

Ubuntu/Debian: sudo systemctl enable nginx

Centos/RHEL: systemctl enable nginx or service nginx enable

Ubuntu/Debian: sudo systemctl start nginx

Centos/RHEL: systemctl start nginx or service start enable

Ubuntu/Debian: sudo systemctl status nginx  
Centos/RHEL: systemctl status nginx or service status enable

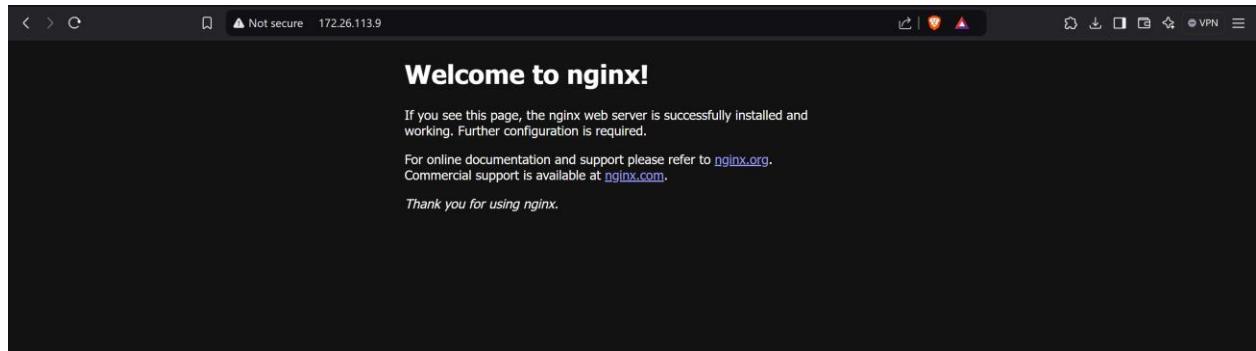
```
mujeeb@Mujeeb:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
mujeeb@Mujeeb:~$ sudo systemctl start nginx
mujeeb@Mujeeb:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-11-24 11:36:38 PKT; 1min 36s ago
       Docs: man:nginx(8)
   Main PID: 3203 (nginx)
      Tasks: 9 (limit: 4612)
     Memory: 6.2M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/nginx.service
           └─3203 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─3204 "nginx: worker process"
             ├─3205 "nginx: worker process"
             ├─3206 "nginx: worker process"
             ├─3207 "nginx: worker process"
             ├─3208 "nginx: worker process"
             ├─3210 "nginx: worker process"
             ├─3211 "nginx: worker process"
             └─3212 "nginx: worker process"

Nov 24 11:36:38 Mujeeb systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Nov 24 11:36:38 Mujeeb systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

- Step 4: You can verify by either accessing port 80 of the machine using it's IP in a web browser. Additionally you can use the “curl” command to check if it returns a web page.

```
curl http://your-machine-ip:80/
```

```
mujeeb@Mujeeb:~$ curl http://172.26.113.9:80/
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```



## Installation of MySQL and creating empty database

- Step 1: Installing mysql server on linux machine

Ubuntu/Debian: sudo apt install mysql-server

Centos/RHEL: sudo dnf install mysql-server

```
mujeeb@mujeeb:~$ sudo apt install mysql-server
[sudo] password for mujeeb:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaio1i64 libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64
  libfcgi-bin libfcgi-perl libfcgi0i64 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediary-perl libmecab2 libnuma1 libprotobuf-lite32t64
  libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl
  libwww-perl mailx tinyca
The following NEW packages will be installed:
  libaio1i64 libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64
  libfcgi-bin libfcgi-perl libfcgi0i64 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediary-perl libmecab2 libnuma1 libprotobuf-lite32t64
  libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 30 newly installed, 0 to remove and 0 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
0% [Working]
```

- Step 2: After installation, you can start the MySQL service using either of the two commands

systemctl start mysql

Or

service mysql start

```
root@e5d965cc7738:/# service mysql start
 * Starting MySQL database server mysqld
   ...failed!
root@e5d965cc7738:/# su: warning: cannot change directory to /no
[ OK ]
root@e5d965cc7738:/# service mysql status
  * /usr/bin/mysqladmin Ver 8.0.40-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))
    Copyright (c) 2000, 2024, Oracle and/or its affiliates.

  Oracle is a registered trademark of Oracle Corporation and/or its
  affiliates. Other names may be trademarks of their respective
  owners.

  Server version      8.0.40-0ubuntu0.24.04.1
  Protocol version   10
  Connection          Localhost via UNIX socket
  UNIX socket        /var/run/mysqld/mysqld.sock
  Uptime:             10 sec

Threads: 2  Questions: 8  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.800
```

- Step 3: The status of the MySQL can be checked using the command

```
~~~~~  
systemctl status mysql
```

or

```
service mysql status  
~~~~~
```

```
root@e54965cc7738:/# service mysql status  
* /usr/bin/mysqladmin Ver 8.0.40-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Server version      8.0.40-0ubuntu0.24.04.1  
Protocol version   10  
Connection          Localhost via UNIX socket  
UNIX socket        /var/run/mysqld/mysqld.sock  
Uptime:             2 min 7 sec  
  
Threads: 2  Questions: 12  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.094
```

- Step 4: Next, we need to log in to MySQL and create a user and initialize an empty database for our application

```
mysql -u root -p
```

-- Database for application

```
CREATE DATABASE your_database_name;
```

-- For remote access

```
CREATE USER 'your_app_user'@'%' IDENTIFIED WITH mysql_native_password BY  
'your_password';
```

-- For remote access

```
GRANT ALL PRIVILEGES ON your_database_name.* TO 'your_app_user'@'%';
```

-- Apply privileges

```
FLUSH PRIVILEGES;  
~~~~~
```

-- Verify

```
SHOW DATABASES;
```

```
SELECT user,host FROM mysql.user;
```

```
~~~~~
```

```

root@SU965cc773B:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database lic_db
->;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER 'lic_user'@'%' IDENTIFIED WITH mysql_native_password BY 'admin';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON lic_db.* TO 'lic_user'@'%';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| lic_db             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql> SELECT user,host FROM mysql.user;
+-----+-----+
| user | host  |
+-----+-----+
| admin | %    |
| lic_user | %    |
| debian-sys-maint | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys | localhost |
| root | localhost |
+-----+-----+
7 rows in set (0.00 sec)

mysql> |

```

- Step 5: Configure Remote Access to allow the remote connection

~~~~~  
*# Edit MySQL configuration*  
 sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

*# Find and change bind-address line to:*  
 bind-address = 0.0.0.0  
*# Or comment it out with*  
*# for all interfaces*

*# Save and restart MySQL*  
 sudo systemctl restart mysql OR service mysql restart  
 ~~~~~

```
root@e54965cc7738:/# nano /etc/mysql/mysql.conf.d/mysqld.cnf
root@e54965cc7738:/# cat /etc/mysql/mysql.conf.d/mysqld.cnf | head -35
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid-file     = /var/run/mysqld/mysqld.pid
# socket       = /var/run/mysqld/mysqld.sock
# port         = 3306
# datadir      = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir        = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address    = 127.0.0.1
bind-address    = 0.0.0.0
mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
root@e54965cc7738:/# service mysql restart
* Stopping MySQL database server mysqld
* Starting MySQL database server mysqld
su: warning: cannot change directory to /nonexistent: No such file or directory
[ OK ]
root@e54965cc7738:/# |
[ OK ]
```

```
root@e54965cc7738:/# nano 7.2
GNU nano 7.2
/etc/mysql/mysql.conf.d/mysqld.cnf *
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid-file     = /var/run/mysqld/mysqld.pid
# socket       = /var/run/mysqld/mysqld.sock
# port         = 3306
# datadir      = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir        = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address    = 127.0.0.1
bind-address    = 0.0.0.0
mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
#
```

Screenshot of file /etc/mysql/mysql.conf.d/mysqld.cnf

With this the installation of MySQL with its tools are installed and configured. Additionally, we have also initialized a database which we will need later.

## *Installation of node.js*

- Step 1: Updating system packages

~~~~~

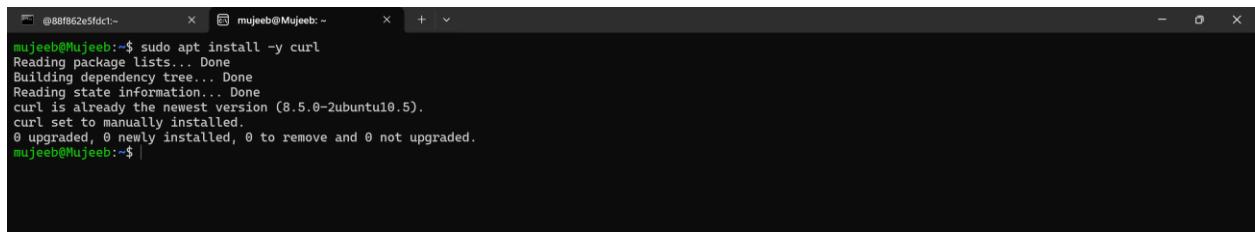
Ubuntu/Debian:

```
sudo apt install curl -y
```

RHEL/CentOS:

```
dnf install curl -y
```

~~~~~



```
mujeeb@Mujeeb:~$ sudo apt install -y curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.5).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
mujeeb@Mujeeb:~$ |
```

- Step 2: Add NodeSource repository for Node.js 20:

~~~~~

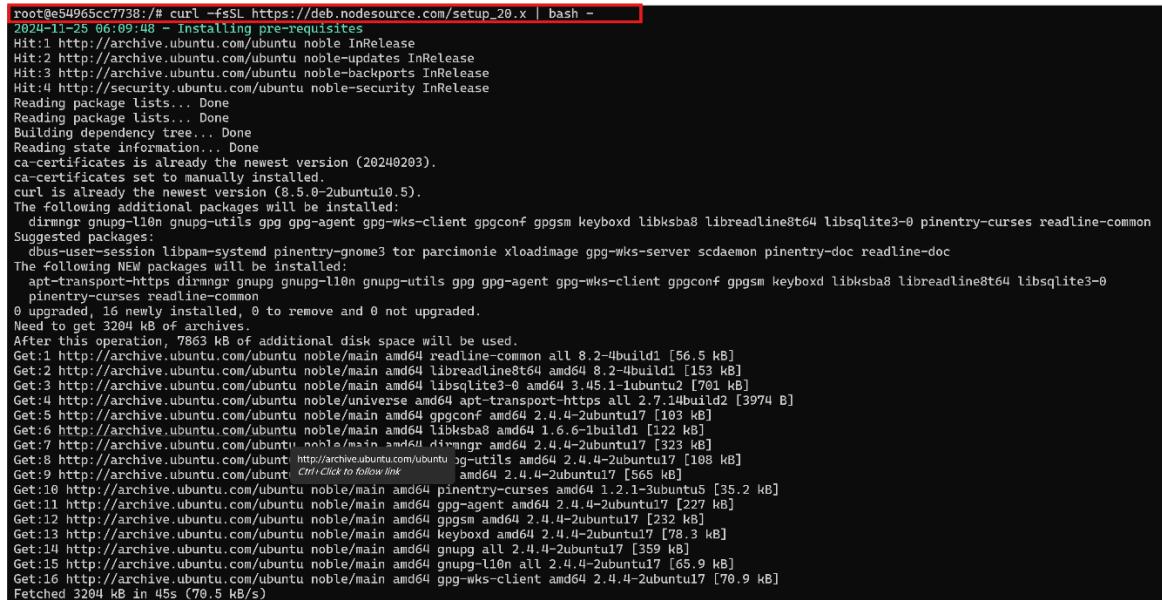
Ubuntu/Debian

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
```

RHEL/CentOS

```
curl -fsSL https://rpm.nodesource.com/setup_20.x | sudo -E bash -
```

~~~~~



```
root@e54965cc7738:/# curl -fsSL https://deb.nodesource.com/setup_20.x | bash -
2024-11-25 06:09:48 - Installing pre-requisites
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.5).
The following additional packages will be installed:
  dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpgconf gpgsm keyboxd libksba8 libreadline8t64 libsqlite3-0 pinentry-curses readline-common
Suggested packages:
  dbus-user-session libpam-systemd pinentry-gnome3 tor parcellite xloadimage gpg-wks-server scdaemon pinentry-doc readline-doc
The following NEW packages will be installed:
  apt-transport-https dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpgconf gpgsm keyboxd libksba8 libreadline8t64 libsqlite3-0
pinentry-curses readline-common
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 3294 kB of archives.
After this operation, 7863 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu/noble/main amd64 readline-common all 8.2-0build1 [56.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu/noble/main amd64 libreadline8t64 amd64 8.2-0build1 [153 kB]
Get:3 http://archive.ubuntu.com/ubuntu/noble/main amd64 libsqlite3-0 amd64 3.45.1-1ubuntu2 [701 kB]
Get:4 http://archive.ubuntu.com/ubuntu/noble/universe amd64 apt-transport-https all 2.7.14+build2 [3974 kB]
Get:5 http://archive.ubuntu.com/ubuntu/noble/main amd64 gpgconf amd64 2.4.4-2ubuntu17 [103 kB]
Get:6 http://archive.ubuntu.com/ubuntu/noble/main amd64 libksba8 amd64 1.6.6-1ubuntu1 [122 kB]
Get:7 http://archive.ubuntu.com/ubuntu/noble/main amd64 dirmngr amd64 2.4.4-2ubuntu17 [323 kB]
Get:8 http://archive.ubuntu.com/ubuntu http://archive.ubuntu.com/ubuntu/gpg-utils amd64 2.4.4-2ubuntu17 [108 kB]
Get:9 http://archive.ubuntu.com/ubuntu gpg-wks-client amd64 2.4.4-2ubuntu17 [565 kB]
Get:10 http://archive.ubuntu.com/ubuntu/noble/main amd64 pinentry-curses amd64 1.2.1-3ubuntu5 [35.2 kB]
Get:11 http://archive.ubuntu.com/ubuntu/noble/main amd64 gpg-agent amd64 2.4.4-2ubuntu17 [227 kB]
Get:12 http://archive.ubuntu.com/ubuntu/noble/main amd64 gpgsm amd64 2.4.4-2ubuntu17 [232 kB]
Get:13 http://archive.ubuntu.com/ubuntu/noble/main amd64 keyboxd amd64 2.4.4-2ubuntu17 [78.3 kB]
Get:14 http://archive.ubuntu.com/ubuntu/noble/main amd64 gnupg-all all 2.4.4-2ubuntu17 [359 kB]
Get:15 http://archive.ubuntu.com/ubuntu/noble/main amd64 gnupg-l10n all 2.4.4-2ubuntu17 [65.9 kB]
Get:16 http://archive.ubuntu.com/ubuntu/noble/main amd64 gpg-wks-client amd64 2.4.4-2ubuntu17 [70.9 kB]
Fetched 3294 kB in 45s (70.5 kB/s)
```

- Step 3: Install Node.js:

```
Ubuntu/Debian: sudo apt install -y nodejs
RHEL/CentOS: sudo dnf install -y nodejs
```

```
root@e54965cc7738:/# apt-get install nodejs -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libexpat1 libpython3-stdlib libpython3.12-minimal libpython3.12-stdlib media-types python3 python3-minimal python3.12 python3.12-minimal tzdata
Suggested packages:
  python3-doc python3-tk python3-venv python3.12-venv python3.12-doc binutils binfmt-support
The following NEW packages will be installed:
  libexpat1 libpython3-stdlib libpython3.12-minimal libpython3.12-stdlib media-types nodejs python3 python3-minimal python3.12 python3.12-minimal tzdata
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 38.1 MB of archives.
After this operation, 223 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-minimal amd64 3.12.3-1ubuntu0.3 [834 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nodejs amd64 20.18.1-1nodesource1 [31.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1 amd64 2.6.1-2ubuntu0.1 [87.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12-minimal amd64 3.12.3-1ubuntu0.3 [2333 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-minimal amd64 3.12.3-0ubuntu2 [27.4 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 media-types all 10.1.0 [27.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 tzdata all 2024a-3ubuntu1.1 [273 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-stdlib amd64 3.12.3-1ubuntu0.3 [2068 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12 amd64 3.12.3-1ubuntu0.3 [651 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3-stdlib amd64 3.12.3-0ubuntu2 [10.0 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3 amd64 3.12.3-0ubuntu2 [23.0 kB]
Fetched 38.1 MB in 40s (942 kB/s)
```

Note: While downloading, it is possible that it prompts to select time zone as shown below

```
Setting up tzdata (2024a-3ubuntu1.1) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 79.)
debconf: falling back to frontend: Readline
Configuring tzdata

Please select the geographic area in which you live. Subsequent configuration questions will narrow this down by presenting a list of cities, representing the time zones in which they are located.

 1. Africa 2. America 3. Antarctica 4. Arctic 5. Asia 6. Atlantic 7. Australia 8. Europe 9. Indian 10. Pacific 11. Etc
Geographic area: 5

Please select the city or region corresponding to your time zone.

 1. Aden    11. Baku    21. Damascus   31. Hong_Kong  41. Kashgar   51. Makassar   61. Pyongyang  71. Srednekolymsk  81. Urumqi
 2. Almaty   12. Bangkok  22. Dhaka     32. Hovd     42. Kathmandu  52. Manila    62. Qatar      72. Taipei     82. Ust-Nera
 3. Amman   13. Barnaul  23. Dili      33. Irkutsk   43. Khandyga  53. Muscat    63. Qostanay   73. Tashkent   83. Vientiane
 4. Anadyr   14. Beirut   24. Dubai     34. Istanbul  44. Kolkata   54. Nicosia    64. Ozylorda  74. Tbilisi    84. Vladivostok
 5. Aqttau   15. Bishkek   25. Dushanbe  35. Jakarta   45. Krasnoyarsk 55. Novokuznetsk 65. Riyadh    75. Tehran    85. Yakutsk
 6. Aqtobe   16. Brunei   26. Famagusta 36. Jayapura  46. Kuala_Lumpur 56. Novosibirsk 66. Salkhalin  76. Tel_Aviv  86. Yangon
 7. Ashgabad 17. Chita    27. Gaza      37. Jerusalem 47. Kuching   57. Omsk      67. Samarkand  77. Thimphu   87. Yekaterinburg
 8. Atyrau   18. Choisalsan 28. Harbin   38. Kabul     48. Kuwait    58. Oral       68. Seoul      78. Tokyo     88. Yerevan
 9. Baghdad   19. Chongqing 29. Hebron   39. Kanchatka 49. Macau     59. Phnom_Penh 69. Shanghai   79. Tomsk
10. Bahrain  20. Colombo  30. Ho_Chi_Minh 40. Karachi   50. Magadan  60. Pontianak  70. Singapore  80. Ulaanbaatar
Time zone: 40

Current default time zone: 'Asia/Karachi'
Local time is now:  Mon Nov 25 11:13:03 PKT 2024.
Universal Time is now: Mon Nov 25 06:13:03 UTC 2024.
Run 'dpkg-reconfigure tzdata' if you wish to change it.

Setting up libpython3.12-stdlib:amd64 (3.12.3-1ubuntu0.3) ...
Setting up python3.12 (3.12.3-1ubuntu0.3) ...
Setting up libpython3-stdlib:amd64 (3.12.3-0ubuntu2) ...
Setting up python3 (3.12.3-0ubuntu2) ...
running python rtpdate hooks for python3.12...
running python post-rtpdate hooks for python3.12...
Setting up nodejs (20.18.1-1nodesource1) ...
Processing triggers for libc-bin (2.39-0ubuntu0.3) ...
```

- Step 4: Verify installation:

```
node --version
npm --version
```

```
root@e54965cc7738:/# node --version
v20.18.1
root@e54965cc7738:/#
root@e54965cc7738:/#
root@e54965cc7738:/# npm --version
10.8.7
root@e54965cc7738:/#
root@e54965cc7738:/#
root@e54965cc7738:/|
```

With this the installation of node.js is completed.

## Download codebase

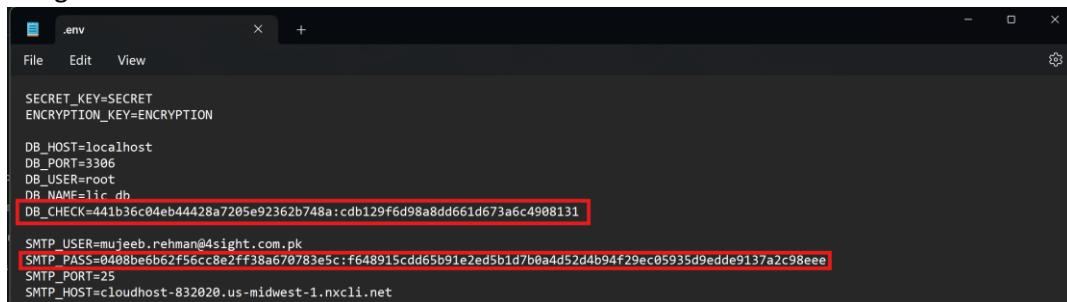
- Now that the dependencies based on your platform are installed, we can move to the application setup. The application will be provided through either a link from where you can install it or through any other medium the client prefers. The basic directory structure you will see is this. You will see only two folders as follows.

|                         |                     |             |
|-------------------------|---------------------|-------------|
| 📁 LicenseManager        | 11/14/2024 10:23 AM | File folder |
| 📁 licenseManagerBackend | 11/14/2024 1:10 PM  | File folder |

Now that all the tools and application is installed. Let us move towards the initial configuration of the application.

## Backend Setup

- o First let's setup the backend. Navigate to your "licenseManagerBackend" directory. The first thing to do is edit the .env file.



```
SECRET_KEY=SECRET
ENCRYPTION_KEY=ENCRYPTION

DB_HOST=localhost
DB_PORT=3306
DB_USER=root
DB_NAME=lic_db
DB_CHECK=441b36c04eb44428a7205e92362b748a:cdb129f6d98a8dd661d673a6c4908131

SMTP_USER=mujeeb_rehman@4sight.com.pk
SMTP_PASS=0408be6b62f56cc8e2ff38a670783e5c:f648915cdd65b91e2ed5b1d7b0a4d52d4b94f29ec05935d9edde9137a2c98eee
SMTP_PORT=25
SMTP_HOST=1cloudhost-832020.us-midwest-1.nxcli.net
```

The .env file contains the database configuration and the SMTP configuration. You will need to change the values based on your configuration.

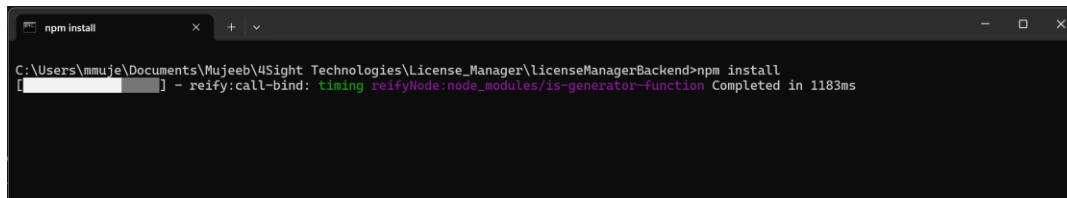
Important Note: Keeping in mind security, you can see that the database password and the SMTP password are both encrypted. Once you enter your password and start the application. The passwords will be encrypted automatically.

- o Second, we need to install all the dependencies of the application. For this reason, open CMD and navigate to the backend directory and run the following command.

~~~~~

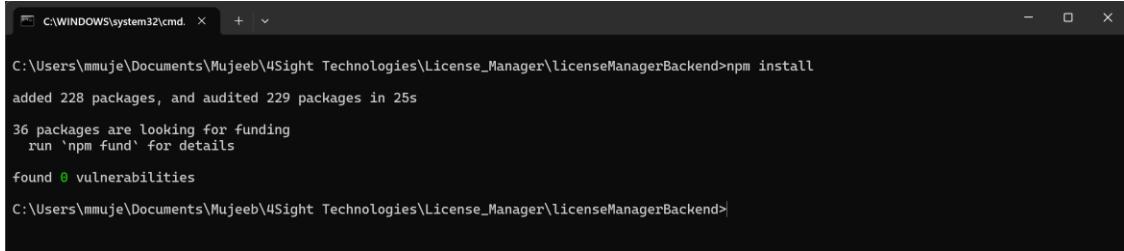
npm install

~~~~~



```
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>npm install
[1/1] - reify:call-bind: timing reifyNode:node_modules/is-generator-function Completed in 1183ms
```

After the installation is complete you should see a message similar to this indicating successful installation of dependencies.



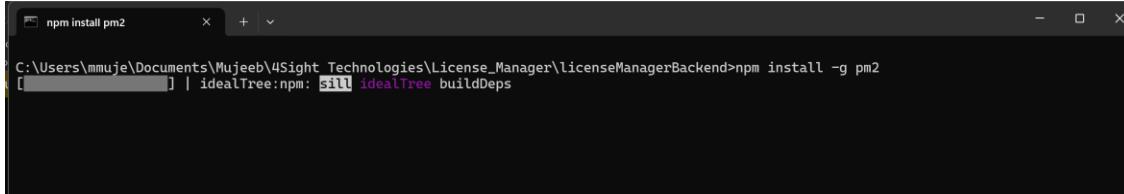
```
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>npm install
added 228 packages, and audited 229 packages in 25s
36 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>
```

- **Important Note:** before proceeding, a license for your application to run is needed to be generated. This step can only be done by the Application Provider.
- Third, after the license generation, now you need to open CMD and navigate to the "licenseManagerBackend" directory and run the following command.

~~~~~

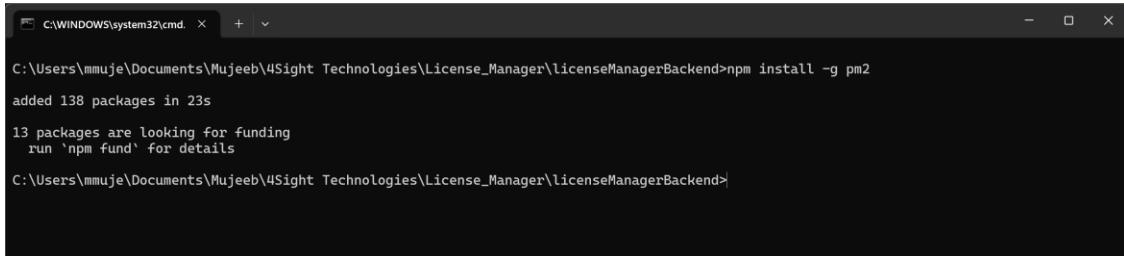
npm install -g pm2

~~~~~



```
npm install pm2
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>npm install -g pm2
[redacted] | idealTree:npm: sill idealTree buildDeps
```

If you do not see such interface, and get an error please check your node installation. After installation you will get this message.



```
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>npm install -g pm2
added 138 packages in 23s
13 packages are looking for funding
  run 'npm fund' for details
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>
```

- Fourth, now our backend is ready to be started using pm2, this is used to ensure the application starts automatically on a reboot. Additionally, pm2 helps in load balancing as well. Run the following commands in CMD staying in the same directory.

~~~~~

pm2 list

~~~~~

This command shows us the list of applications that pm2 is handling. Initially it should be empty.

| id | name                  | namespace | version | mode | pid | uptime | status | cpu | mem | user | watching |
|----|-----------------------|-----------|---------|------|-----|--------|--------|-----|-----|------|----------|
|    | LicenseManagerBackend |           |         |      |     |        | online |     |     |      |          |

~~~~~  
**pm2 start app.js --name LicenseManagerBackend**  
~~~~~

This command starts the application using pm2 service manager. After this you can run the following command to save our state or list of services. This will help us when rebooting the machine.

~~~~~  
**pm2 save**  
~~~~~

| id | name                  | namespace | version | mode | pid   | uptime | status | cpu    | mem | user   | watching |          |
|----|-----------------------|-----------|---------|------|-------|--------|--------|--------|-----|--------|----------|----------|
| 0  | LicenseManagerBackend | default   | 1.0.0   | fork | 10584 | 0s     | 0      | online | 8%  | 38.8mb | mmuje    | disabled |

[PM2] [WARN] Current process list is not synchronized with saved list. Type 'pm2 save' to synchronize.  
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License\_Manager\licenseManagerBackend>pm2 save  
[PM2] Saving current process list...  
[PM2] Successfully saved in C:\Users\mmuje\pm2\dump.pm2  
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License\_Manager\licenseManagerBackend>

- To make sure that your application has started and it has connected to the database. We can run the following command.

~~~~~  
**pm2 log LicenseManagerBackend**  
~~~~~

```
[TAILING] Tailing last 15 lines for [LicenseManagerBackend] process (change the value with --lines option)
C:\Users\mmuje\pm2\logs\LicenseManagerBackend-out.log last 15 lines:
[0|LicenseM | DB_CHECK is already encrypted.
[0|LicenseM | SMTP_PASS is already encrypted.
[0|LicenseM | Server started on http://localhost:5000
[0|LicenseM | Database connection has been established successfully.
||
```

If you see error logs, it is highly likely that there is issue in database configuration. More information can be inferred from the error logs.

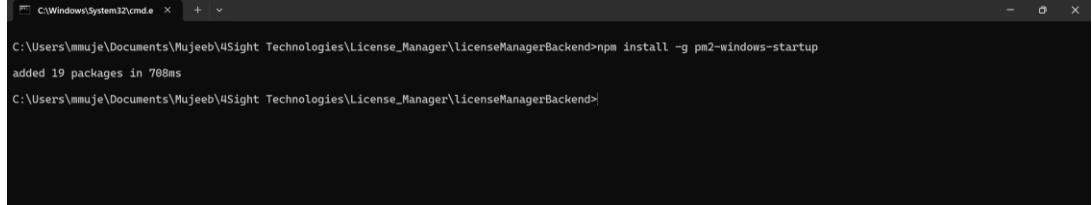
- Now our backend is setup and running. Here we have a difference of setup in linux and windows to ensure our application starts when the machine reboots. Follow the below step based on your Platform.

### Steps for windows:

- o Step 1: In **CMD**, run the following command to install a package for our application

```
npm install -g pm2-windows-startup
```

```
~~~~~
```

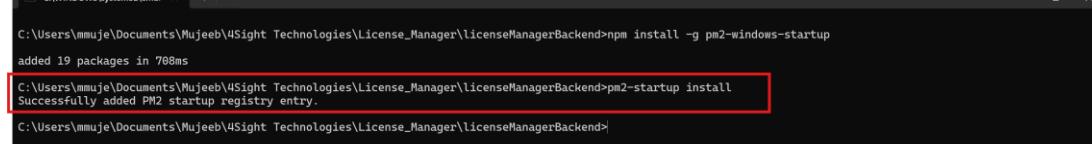


```
C:\Windows\System32\cmd.e > + <
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>npm install -g pm2-windows-startup
added 19 packages in 708ms
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>
```

- o Step 2: Now run the below command to ensure that your application is started if the machine reboots.

```
pm2-startup install
```

```
~~~~~
```

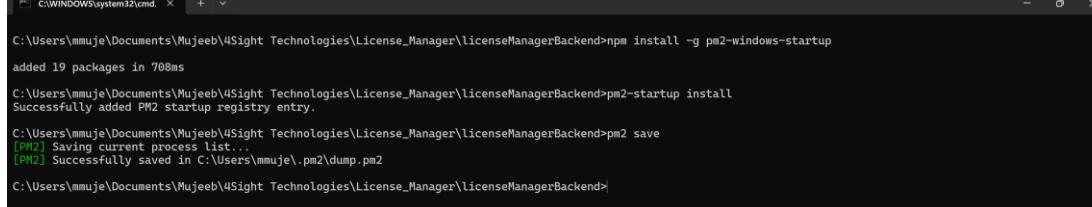


```
C:\Windows\System32\cmd.e > + <
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>npm install -g pm2-windows-startup
added 19 packages in 708ms
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>pm2-startup install
Successfully added PM2 startup registry entry.
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>
```

- o Step 3: Lastly, run the following command to save our service managers state.

```
pm2 save
```

```
~~~~~
```



```
C:\Windows\System32\cmd.e > + <
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>npm install -g pm2-windows-startup
added 19 packages in 708ms
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>pm2-startup install
Successfully added PM2 startup registry entry.
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in C:\Users\mmuje\.pm2\dump.pm
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\licenseManagerBackend>
```

### Steps for Linux:

- o Step 1: On linux it is simpler to achieve this, we do not need to install any other package. After you had checked the log for the application, you need to run the following command.

```
pm2 startup
```

```
~~~~~
```

```
root@5f4965cc7738:/# pm2 startup
[PM2] Init System found: upstart
Platform upstart
Template
#!/bin/bash
### BEGIN INIT INFO
# Provides: pm2
# Required-Start: $local_fs $remote_fs $network
# Required-Stop: $local_fs $remote_fs $network
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: PM2 Init script
# Description: PM2 process manager
### END INIT INFO

NAME=pm2
PM2=/usr/lib/node_modules/pm2/bin/pm2
USER=undefined
DEFAULT=/etc/default/$NAME

export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:$PATH
export PM2_HOME="/root/.pm2"

# The following variables can be overwritten in $DEFAULT

# maximum number of open files
MAX_OPEN_FILES=

# overwrite settings from default file
if [ -f "$DEFAULT" ]; then
    . "$DEFAULT"
fi

# set maximum open files if set
if [ -n "$MAX_OPEN_FILES" ]; then
    ulimit -n $MAX_OPEN_FILES
fi

get_user_shell() {
    local shell
    shell=$(getent passwd "${1:-$(whoami)}" | cut -d: -f7 | sed -e 's/[[:space:]]*$/\'')
    if [[ $shell == /*$/sbin/nologin" ]] || [[ $shell == /*$/bin/false" ]] || [[ -z "$shell" ]]; then
```

```
;;
force-reload)
    reload
;;
*)
    echo "Usage: {start|stop|status|restart|reload|force-reload"
    exit 1
;;
esac
exit $RETVAL

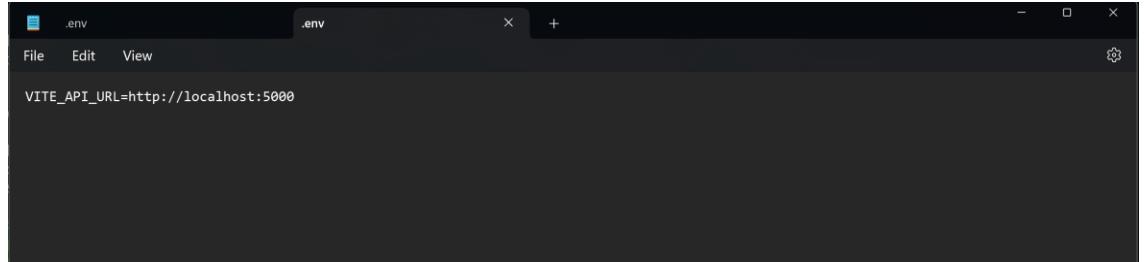
Target path
/etc/init.d/pm2-undefined
Command list
[
    'chmod +x /etc/init.d/pm2-undefined',
    'mkdir -p /var/lock/subsys',
    'touch /var/lock/subsys/pm2-undefined',
    'update-rc.d pm2-undefined defaults'
]
[PM2] Writing init configuration in /etc/init.d/pm2-undefined
[PM2] Making script booting at startup...
[PM2] [-] Executing: chmod +x /etc/init.d/pm2-undefined...
[PM2] [v] Command successfully executed
[PM2] [-] Executing: mkdir -p /var/lock/subsys...
[PM2] [v] Command successfully executed.
[PM2] [-] Executing: touch /var/lock/subsys/pm2-undefined...
[PM2] [v] Command successfully executed.
[PM2] [-] Executing: update-rc.d pm2-undefined defaults...
[PM2] [v] Command successfully executed.
+-----+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup upstart
```

And that is the only step. This will ensure that the application starts automatically if the machine reboots. Do keep in mind to run “**pm2 save**” at the end to ensure that all settings are saved.

## Frontend Setup

- Step 1: now we need to setup the frontend. Navigate to the “licenseManager” directory and open the .env file. Add your backend application url here.



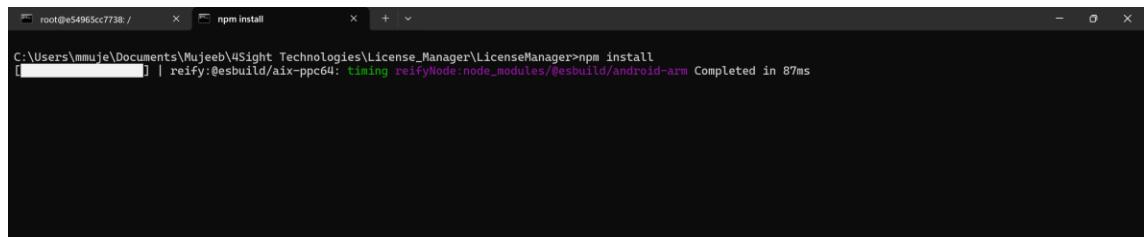
```
VITE_API_URL=http://localhost:5000
```

- Step 2: open “CMD” and navigate to the “licenseManager” directory and download all the dependencies.

~~~~~

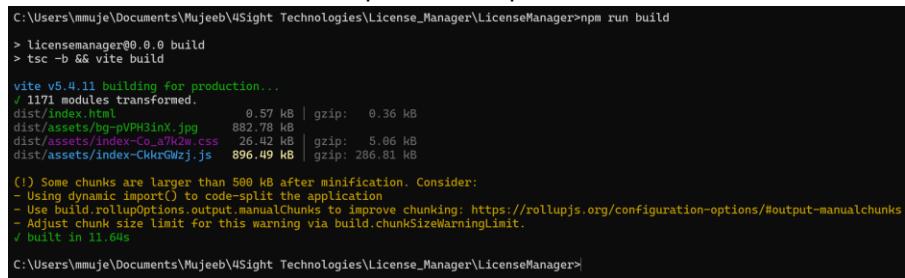
npm install

~~~~~



```
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager>npm install
[...]
[reify:@esbuild/aix-ppc64: timing reifyNode@node_modules/@esbuild/android-arm Completed in 87ms]
```

- Step 3: Now we need to build our application which then will be hosted on the web server we had installed in the previous steps.



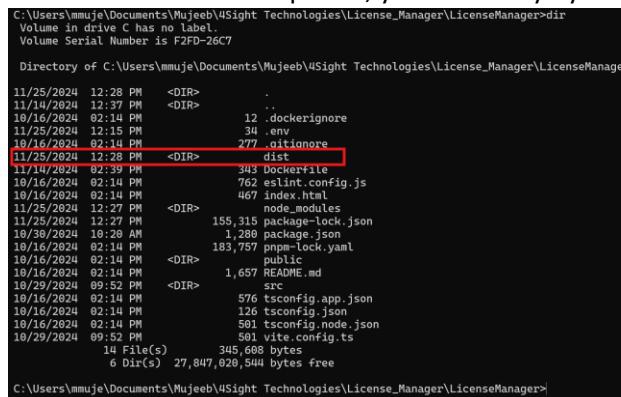
```
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager>npm run build
> licensemanager@0.0.0 build
> tsc -b && vite build

vite v5.0.11 building for production...
/ 1171 modules transformed.
dist/index.html          0.57 kB | gzip:  0.36 kB
dist/assets/bg-pPH3inX.jpg 882.78 kB
dist/assets/index-Co_aTK2W.css 26.42 kB | gzip:  5.86 kB
dist/assets/index-CkkcGWzj.js 896.49 kB | gzip: 286.81 kB

(!) Some chunks are larger than 500 kB after minification. Consider:
- Using dynamic import() to code-split the application
- Use build.rollupOptions.output.manualChunks to improve chunking: https://rollupjs.org/configuration-options/#output-manualchunks
- Adjust chunk size limit for this warning via build.chunkSizeWarningLimit.
/ built in 11.64s

C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager>
```

Note: After build is completed, you can verify by checking if you have a “dist” folder.



```
C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager>dir
Volume in drive C has no label
Volume Serial Number is F2FD-26C7

Directory of C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\LicenseManager

11/25/2024 12:28 PM <DIR> .
11/14/2024 12:37 PM <DIR> ..
10/16/2024 02:14 PM 12 .dockerignore
10/15/2024 02:14 PM 34 .env
10/15/2024 02:14 PM 277 .gitignore
11/25/2024 12:28 PM <DIR> dist
11/14/2024 02:39 PM 343 Dockerfile
10/16/2024 02:14 PM 762 eslint.config.js
10/16/2024 02:14 PM 467 index.html
11/25/2024 12:27 PM <DIR> node_modules
11/25/2024 12:27 PM 155,315 package-lock.json
10/30/2024 10:20 AM 1,280 package.json
10/16/2024 02:14 PM 183,757 pnp-lock.yaml
10/16/2024 02:14 PM <DIR> public
10/16/2024 02:14 PM 1,657 README.md
10/29/2024 09:52 PM <DIR> src
10/16/2024 02:14 PM 576 tsconfig.app.json
10/16/2024 02:14 PM 126 tsconfig.json
10/16/2024 02:14 PM 501 tsconfig.node.json
10/29/2024 09:52 PM 501 vite.config.ts
14 File(s) 345,668 bytes
6 Dir(s) 27,847,020,544 bytes free

C:\Users\mmuje\Documents\Mujeeb\4Sight Technologies\License_Manager\LicenseManager>
```

- Step 4: From this point, the deployment for Linux and Windows is different. In the below steps, the guide shows how to deploy the site on IIS in windows and on nginx for linux.

Steps for windows:

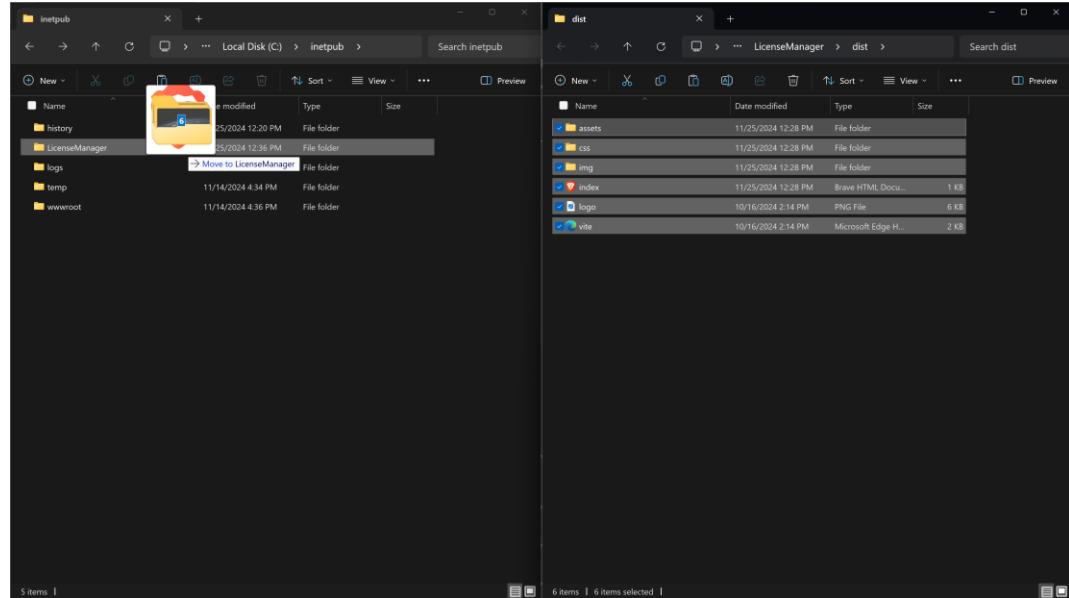
- Step 1: To ensure that there are no permission conflicts, I placed my application files in the default directory of IIS.

~~~~~

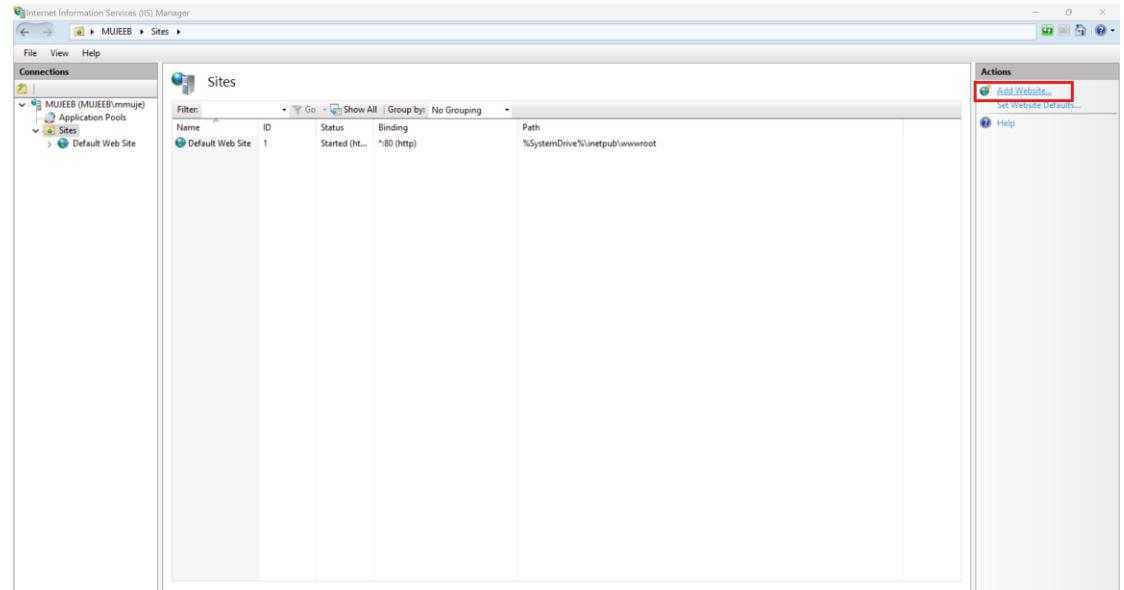
Default directory: C:\inetpub

~~~~~

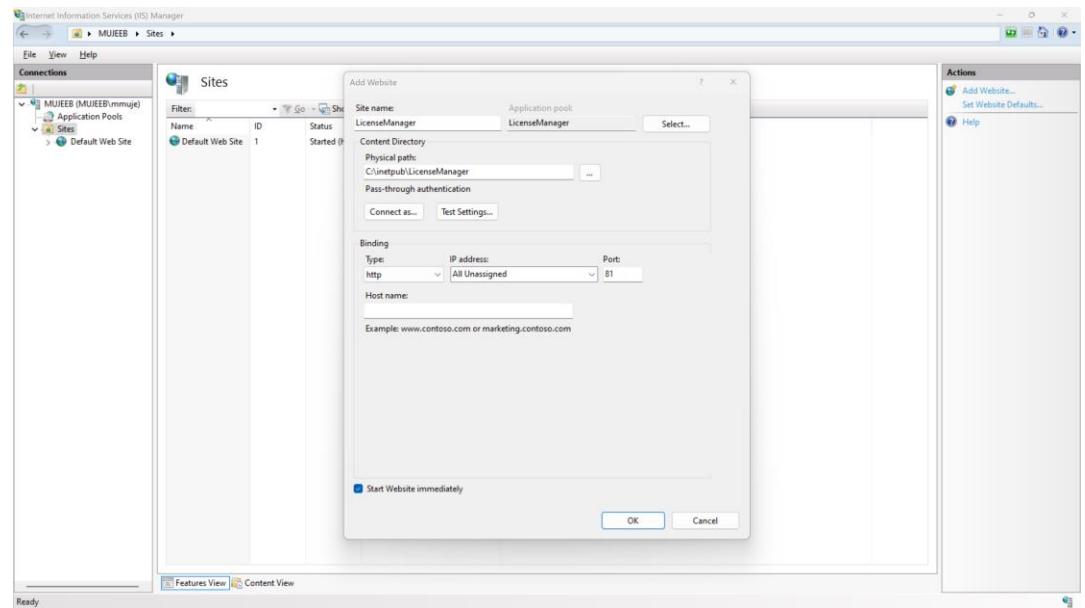
Create a new folder for your application and copy the contents from the “dist” directory into the new directory you created.



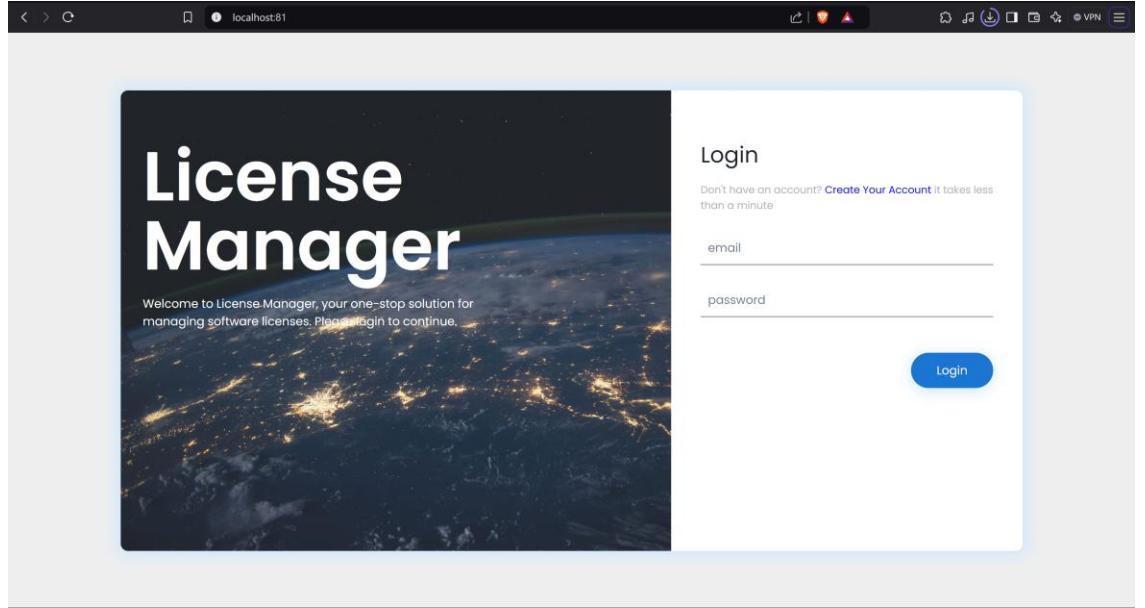
- Step 2: Open your IIS and go to sites section and click on Add Website to host our frontend application.



- Step 3: Adding our website to IIS
  - 1. Add a name for your website and leave the Application pool to what it is.
  - 2. Select the path where the files are placed. As per our step 1, we have placed them into C:\inetpub\LicenseManager.
  - 3. Lastly, you can change the port, I myself have changed the port to 81.
  - 4. Click "OK"



- Step 4: Last step is to verify our deployment. Now we can access our application through the web browser.



#### Steps for Linux:

- Step 1: Verify that you have nginx installed and you have the application on the machine. If anything is missing, please follow the steps in the previous sections.

```
[root@192.168.111.129 main ~]$ ls -la | grep -i "LicenseManager"
drwxr-xr-x. 5 root root 86 Nov 25 18:42 LicenseManager
drwxr-xr-x. 2 root root 6 Nov 24 15:44 licenseManagerBackend
[root@192.168.111.129 main ~]$ systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-11-25 18:49:05 +04; 4min 29s ago
     Process: 4126 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Process: 4124 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 4122 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
   Main PID: 4128 (nginx)
      Group: /system.slice/nginx.service
         ├─4128 nginx: master process /usr/sbin/nginx
         └─4129 nginx: worker process

Nov 25 18:49:05 192.168.111.129main systemd[1]: Stopped The nginx HTTP and reverse proxy server.
Nov 25 18:49:05 192.168.111.129main systemd[1]: Starting The nginx HTTP and reverse proxy server...
Nov 25 18:49:05 192.168.111.129main nginx[4124]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Nov 25 18:49:05 192.168.111.129main nginx[4124]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Nov 25 18:49:05 192.168.111.129main systemd[1]: Started The nginx HTTP and reverse proxy server.
[root@192.168.111.129 main ~]$
```

- Step 2: In the LicenseManager directory I placed the contents of the “dist” folder.

```
[root@192.168.111.129 main ~]$ ls -la LicenseManager/
total 20
drwxr-xr-x. 5 root root 86 Nov 25 18:42 .
drwxr-xr-x-. 8 root root 4096 Nov 25 18:41 ..
drwxr-xr-x. 2 root root 77 Nov 25 18:42 assets
drwxr-xr-x. 2 root root 22 Nov 25 18:42 css
drwxr-xr-x. 2 root root 19 Nov 25 18:42 img
-rw-r--r--. 1 root root 565 Nov 25 11:28 index.html
-rw-r--r--. 1 root root 5559 Oct 16 13:14 logo.png
-rw-r--r--. 1 root root 1497 Oct 16 13:14 vite.svg
[root@192.168.111.129 main ~]$
```

- Step 3: Next, navigate to the directory “/usr/share/nginx/html/”

```
cd /usr/share/nginx/html/
```

```
rm -rf *
```

```
[root@192.168.111.129 main] $cd /usr/share/nginx/html
[root@192.168.111.129 main html] $ls
004.html 50x.html en-US icons img index.html nginx-logo.png poweredby.png
[root@192.168.111.129 main html] $rm -rf *
[root@192.168.111.129 main html] $ls
[root@192.168.111.129 main html] $
```



- Step 4: Now we need to copy the files from our “LicenseManager” directory to this “html” directory.

```
cp -r path-to-LicenseManager/* /usr/share/nginx/html/
systemctl restart nginx
```

```
[root@192.168.111.129 main html] $cp -r ~/LicenseManager/* /usr/share/nginx/html/
[root@192.168.111.129 main html] $ls
assets css img index.html logo.png vite.svg
[root@192.168.111.129 main html] $systemctl restart nginx
[root@192.168.111.129 main html] $
```

- Step 5: Now we can access our application in a web browser

