

# Configuration Manual

Ragul Murugesan

Student ID: 23154870

## 1. Introduction

This document provides detailed instruction about the components that are needed, and respective configuration that needs to be done in order to recreate the project environment in another machine. The project has to install node modules that helps to run the web application. This can we run on any platform as long as node runtime is installed with its respective packages. Detailed follow through installations will be continued below.

## 2. Hardware Machine Configuration

The physical machine on which setup was done is a Techno Megabook T1 with the following specifications.

System Configuration	
Operating System	Microsoft Windows 11 Home Single Edition 64bit
Processor	Intel i7 11 <sup>th</sup> Gen – 1165G7 @ 2.80GHz, Logical Procressor:8, cores: 4
Ram	16GB
Virtual Memory	9.45GB
Graphics Card	4GB
Virtualization	Enabled at Bios

## 3. Software requirements:

The upcoming stages will be discussed about the installation and configuration of software components needed for this project.

- NodeJs: v23.3.0[1]

Node.js is not dependent on any operating system software. It can work on Linux, macOS, or Windows. Node.js is a JavaScript runtime environment with community packages with unique functionality respectively.

[NodeJs installation guide](#) from their respective official website.

- Google Chrome: v 133.0.6876.4

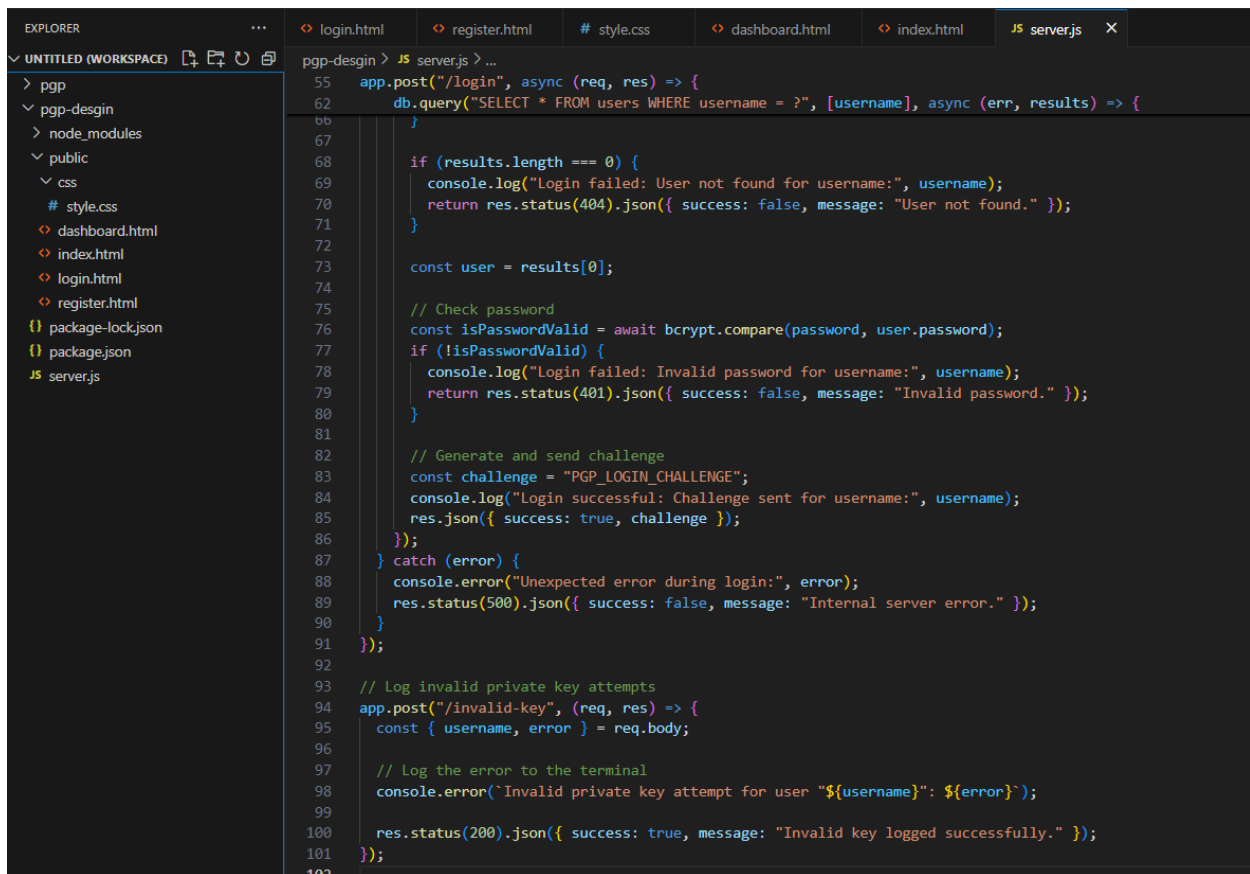
Browser to view the running application with url: localhost.

- Visual Studio Code: v 1.95.3[2]

Code editor to view source code to edits, implements and updates.

- MySQL: v 8.0.41[3]

Used to store data of user information and public key when registration.



The screenshot shows the Visual Studio Code IDE with the 'pgp-desgin' project open. The Explorer sidebar on the left shows the project structure: 'pgp' (parent), 'pgp-desgin' (current), 'node\_modules', 'public', 'css' (containing 'style.css'), 'dashboard.html', 'index.html', 'login.html', 'register.html', 'package-lock.json', and 'package.json'. The main editor displays the 'server.js' file with the following JavaScript code:

```
pgp-desgin > JS server.js > ...
55 app.post("/login", async (req, res) => {
62   db.query("SELECT * FROM users WHERE username = ?", [username], async (err, results) => {
66     }
67   }
68   if (results.length === 0) {
69     console.log("Login failed: User not found for username:", username);
70     return res.status(404).json({ success: false, message: "User not found." });
71   }
72   const user = results[0];
73
74   // Check password
75   const isValidPassword = await bcrypt.compare(password, user.password);
76   if (!isValidPassword) {
77     console.log("Login failed: Invalid password for username:", username);
78     return res.status(401).json({ success: false, message: "Invalid password." });
79   }
80
81   // Generate and send challenge
82   const challenge = "PGP_LOGIN_CHALLENGE";
83   console.log("Login successful: Challenge sent for username:", username);
84   res.json({ success: true, challenge });
85 });
86 } catch (error) {
87   console.error("Unexpected error during login:", error);
88   res.status(500).json({ success: false, message: "Internal server error." });
89 }
90 });
91
92 // Log invalid private key attempts
93 app.post("/invalid-key", (req, res) => {
94   const { username, error } = req.body;
95
96   // Log the error to the terminal
97   console.error(`Invalid private key attempt for user "${username}": ${error}`);
98
99   res.status(200).json({ success: true, message: "Invalid key logged successfully." });
100 });
101
102
```

3.1 Javascript, with servercode in Visual Studio code IDE.

## 4. Installed Package Dependencies:

The below mentioned packages are basic ones to get the application running and its mandatory so it can be installed “npm install bcryptjs body-parser express mysql2 openpgp” or just “npm install” will install the dependencies from projects package.json.

- bcryptjs: ^2.4.3
- body-parser: ^1.20.3
- express: ^4.21.1
- mysql2: ^3.11.3
- openpgp: ^5.11.2[4]

```
const express = require("express");
const bcrypt = require("bcryptjs");
const mysql = require("mysql2");
const bodyParser = require("body-parser");
const path = require("path");
const openpgp = require("openpgp");
```

4.1 Importing packages into server.js

## 5. Connecting SeverJS to Mysql Database:

Step1: Need to change values of user, password, database(schema) to your needs.

```
// Set up MySQL database connection
const db = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "pgp"
});
```

5.1 Change the user, password, database information

Step2: create Database and Tables with below query.

```
CREATE SCHEMA pgp;

CREATE TABLE pgp.users (
  username VARCHAR(255) NOT NULL,
  email VARCHAR(255) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL,
  public_key TEXT NOT NULL,
  PRIMARY KEY (username)
);
```

## 6. Execution and Testing:

Execution and testing phase consists of the following

Phase1: Run “node server.js” to get the application running then enter localhost:3000 on chrome browser.

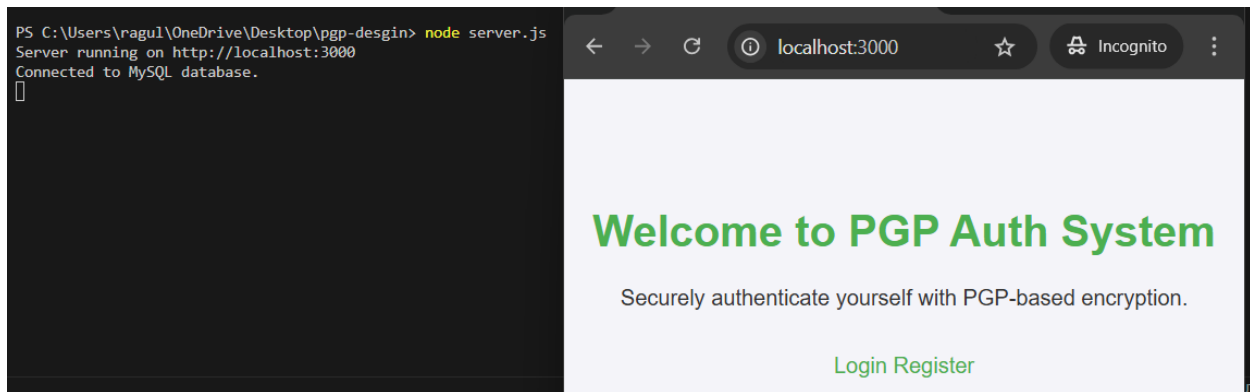


Figure 6.1 Start Sever

Phase2: Registration Page, user will enter the following details shown in the browser once that's done click register button and a unique private\_key will get downloaded in the users system.

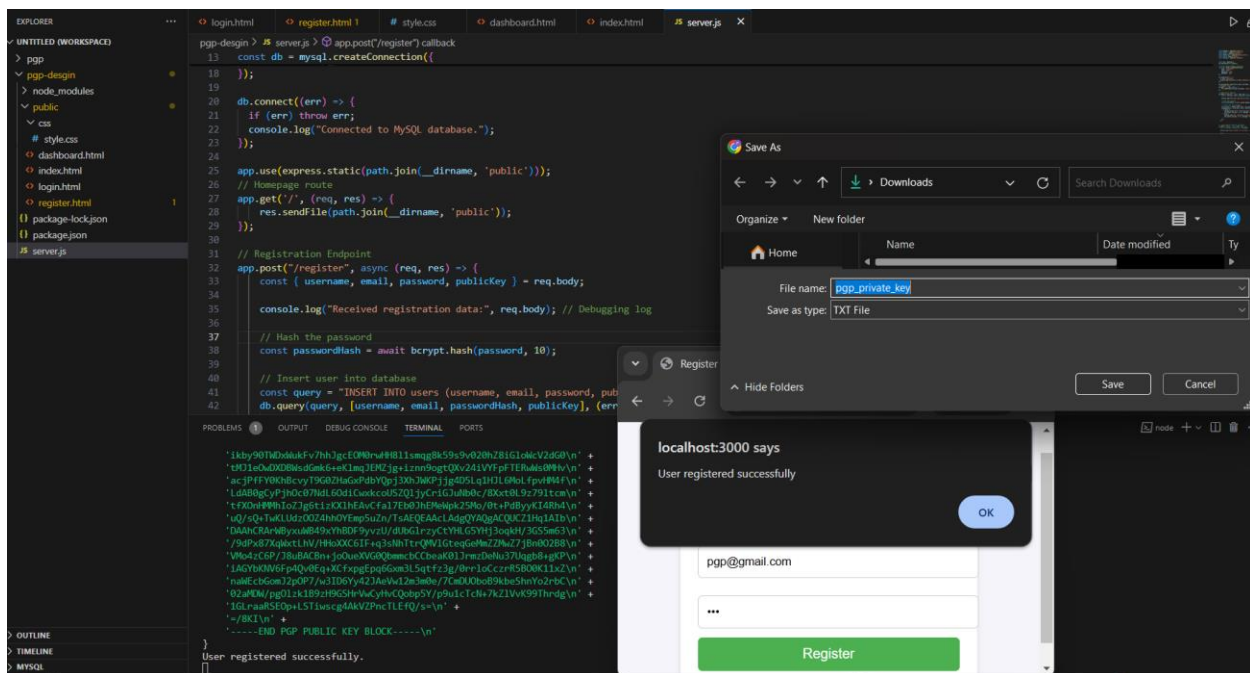


Figure 6.2 User registration and Private\_Key Generation

id	username	email	password	public_key
1	a	a@a.a	\$2a\$10\$KnPikcISp63U.GkhGw94u.OEkbnaP0Jf...	-----BEGIN PGP PUBLIC KEY BLOCK----- xsBNBG...
2	test	test@test.com	\$2a\$10\$RX8gGLKYE6mToQiInxcaA.7oC/Vq6v5...	-----BEGIN PGP PUBLIC KEY BLOCK----- xsBNBG...
3	pgp	pgp@gmail.com	\$2a\$10\$GqLa1U87u1xvGhBztdxbus5VT22QRD...	-----BEGIN PGP PUBLIC KEY BLOCK----- xsBNBG...
4	NULL	NULL	NULL	NULL

Figure 6.3 Check the database if the information with public\_key as been stored.

Phase3: Login page, user will enter their respective username and password if found in database then it creates a security challenge to verify its private key if it matches the public key to get approved by the server.

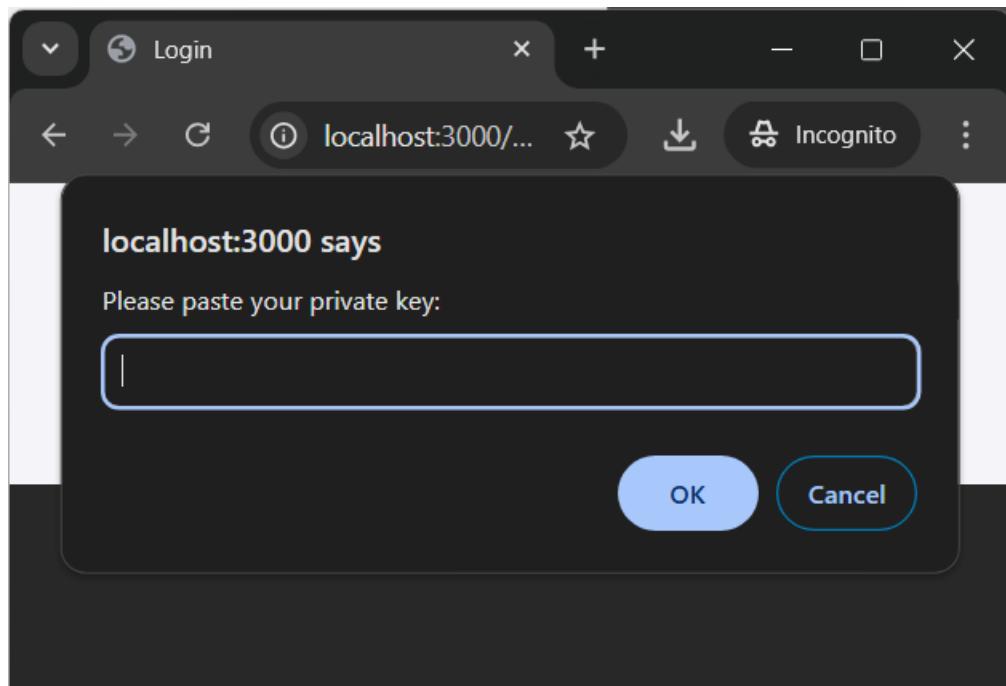


Figure 6.4 Copy paste the Private\_key that was downloaded upon successful registration.

```
-----BEGIN PGP PRIVATE KEY BLOCK-----
xCMGBGdR6tQBCADkbc10zmp8NSOX+wicZkJ7EPb00Ep4GpDM1bUT7iAGRHno
1IaRUW/XfyoYJhJAginE53tZBjcXGYIW2l2eqZI0PRAU2YE5T0duUuAbid95
3SSvM7Kd2bl58mULhrIvcMsbPnN5uSExb9F9UCrzm8RtFhBltiY4LPBBxei
9Vj77x0qk5p0A95ynIzplE0i8gLiRzvjeMpcY8+s7kycsngY0B4iCbczi0nw
4b8Sb7hywb8xj5GLVDvGnjJE3+VsVEfyYcQFXe0yH+wltIEdkpWwY4VbZ1yH
YPduvIwi3rtJJxfKXocVzgyqKTcEZqRVPY9/u8J+PRLKSNkBZba9v1g7ABEB
AAH+CQMI/golQDQBeLzGEtu9qDdSmm1GHxN5mABSwt8+oks1Vw31h1nAXX
KOILtzbGhs6acMfgBP+CW/L7WDYWiQGcgGFGww89DxdTEhkrcFGTNjZ1hlsl
huF2IabCZfSYJVPFq4zNtJeIemG5dnX/4EDzywhaG3D/kZS56fmWgeQWzQH5
dcFY6nTKC0pSSerhoFGYmpw4BqSZ61qLqSUFa0YuZ27cG7STGnII00vbDf4F
GeYH99G+nUpZ/nuEqEH+XfYjvpVx17lCpq5TkCs7lBCVMRYRLutjQzaSuvmm
iImWDrF7/Ueq3Qg9aq4awfsCtgDup9BMA2aaHMjmdURTP+1Cx5JMZdhh+YZC
im9I7HgrU17yQG6MdfYwy6jZg88AiUAF6wT0td1XKuKStStd5HVmgKpjzbQS
6ha/XtEJJjKSK0RnQoVIRHhzsPYv/gJ0EhaujdUUI2Df1s035igDZ5rp2mdu
48bkawj5ZUdcPzRQtPMWqhSPjeKpYERk2qGdbGf95R1X0+5ZFbafQUGSotl
B31Qs7uiNqnk+ZFYULcqsGolAXpQYEkWfX9gsK/Xgryhwdq0+GdexT2oPshG
08RyYEEs3zMSYBXvW2zFve+gPMO6rbfxAUiVSo+LYFo46o6C+Mh8H7HcIUxd
iN+PN0vdtTw0812JutMH7lRm6lwBwO6Vy6XvY1+GJLGE1xkTtxcnkw5ZYA8
7gwPAZ2IJWgayQKD1MI35s7Y4r9FXP6YMT02yIyajs1UEv4Ie4nyUnjT3oaY
czLVpAhkM3H6e480nq5G1Pj9oHqBkea4gR6Qrf/Lx5q4h+bKp0NG0IVxdCF
```

Figure 6.5 private\_key

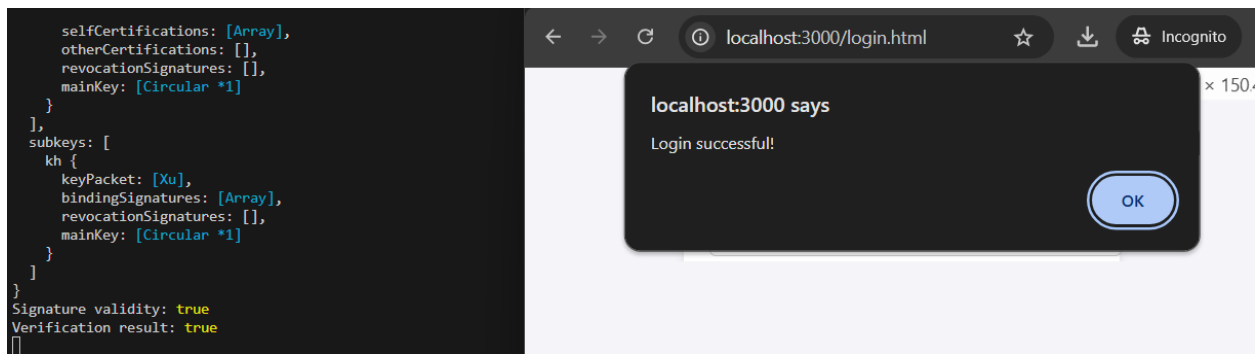


Figure 6.6 Once the Challenge is complete a popup will show Login Successful!

Phase4: The user dashboard upon successful cryptography challenge

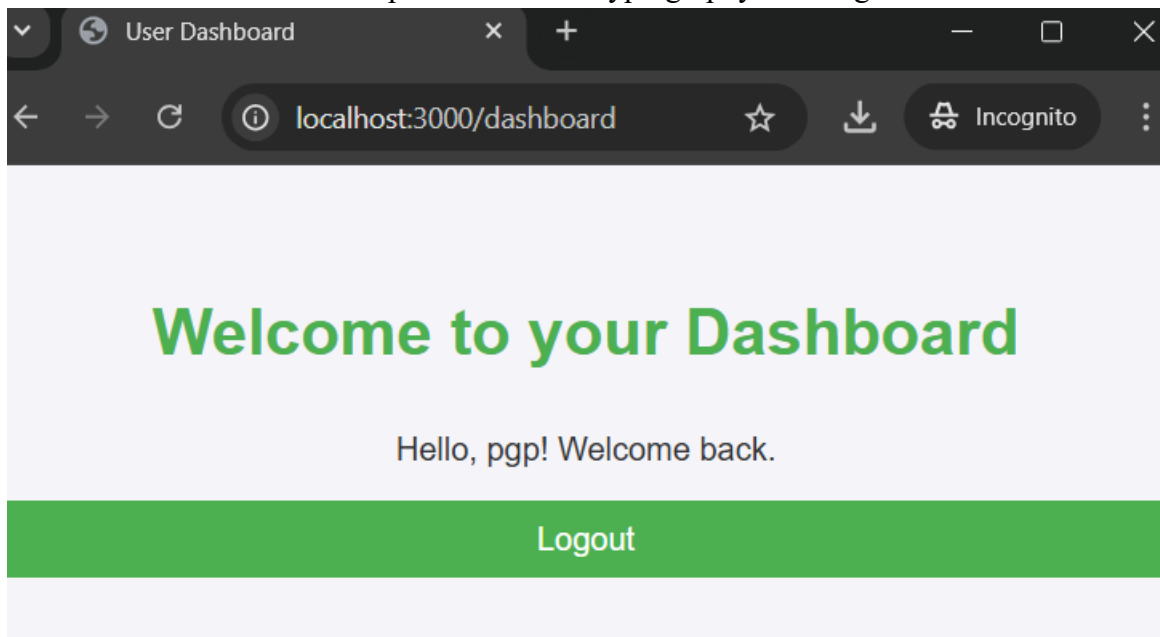


Figure 6.7 Dashboard with logout button.

## References

- [1] "Node.js — Run JavaScript Everywhere." Accessed: Dec. 05, 2024. [Online]. Available: <https://nodejs.org/en>
- [2] "Visual Studio Code - Code Editing. Redefined." Accessed: Dec. 05, 2024. [Online]. Available: <https://code.visualstudio.com/>
- [3] "MySQL :: MySQL Workbench." Accessed: Dec. 05, 2024. [Online]. Available: <https://www.mysql.com/products/workbench/>
- [4] "openpgp - npm." Accessed: Dec. 05, 2024. [Online]. Available: <https://www.npmjs.com/package/openpgp>