



DP World Internship – Review

By Praveen K

Day to day progress





DAY 01 – 09/06/2025

- Induction programs – Safety and IT
- Safety induction
- IT induction





Project 1



PDF Chatbot using Gemini Vision AI-powered Document Assistant

Developed an intelligent chatbot that extracts and interacts with content from uploaded PDF documents using Google's Gemini Vision API. This project allows users to ask natural language questions about the PDF, and the bot responds with context-aware answers. Key features include:

- PDF Parsing & Preprocessing
- Gemini Vision Integration
- Frontend-Backend Integration

DP world

Current Role: Admin | Switch to Employee

Admin Panel

Upload Knowledge Base Files

+ Drag and drop files here, or click to select

Select Files

Uploaded Files (0)

No files uploaded yet

Knowledge Base Status

✓ Knowledge base is active and ready

AI Assistant Chat

Logged in as admin

Hello! I'm your AI assistant. How can I help you today?

11:41:48 am

Type your message...

Press Enter to send, Shift+Enter for new line

DP world

Current Role: Employee | Switch to Admin

AI Assistant Chat

Logged in as employee

Hello! I'm your AI assistant. How can I help you today?

11:41:48 am

Type your message...

Press Enter to send, Shift+Enter for new line

File Edit Selection View Go Run Terminal Help

DIRECTORY: D:\D.P.WORLD.PRAVEEN

File Explorer

```
project1 > frontend > src > components > AdminPanel.jsx >
1 import React, { useState } from "react";
2
3 export default function AdminPanel() {
4   const [uploadedfiles, setuploadedfiles] = useState([]);
5   const [isDragging, setIsDragging] = useState(false);
6
7   const handlefileupload = (files) => {
8     if (!files) return;
9
10    Array.from(files).forEach((file) => {
11      const newfile = {
12        id: Math.random().toString(36).substr(2, 9),
13        name: file.name,
14        size: file.size,
15        uploadedat: new Date(),
16      };
17
18      setuploadedfiles((prev) => [...prev, newfile]);
19      alert(`${file.name} has been uploaded to the knowledge base.`);
20    });
21  };
22
23  const handleDrop = (e) => {
24    e.preventDefault();
25    setIsDragging(true);
26    handlefileupload(e.dataTransfer.files);
27  };
28
29  const handleDragOver = (e) => {
30    e.preventDefault();
31    setIsDragging(true);
32  };
33
34  const handleDragLeave = (e) => {
35    e.preventDefault();
36    setIsDragging(false);
37  };

```

Live Share Fetching data for better TypeScript Intellisense

File Explorer

- project1
- backend
- frontend
- node_modules
- public
- src
- assets
- components
- AdminPanel.jsx
- ChatInterface.jsx
- TypingIndicator.jsx
- contexts
- App.css
- App.jsx
- index.css
- main.jsx
- .gitignore
- .eslintrc.json
- .index.html
- .package-lock.json
- .package.json
- README.md
- vite.config.js
- jp.resume.pdf
- old.py
- resume_praeven.pdf
- RohitResume.pdf
- project2
- project3
- OUTLINE
- TIMELINE

Line 1, Col 1 Spaces: 2 UTF-8 CRLF JavaScript JSX

ENG IN 11:47 07-07-2025

File Edit Selection View Go Run Terminal Help

DIRECTORY: D:\D.P.WORLD.PRAVEEN

File Explorer

```
project1 > frontend > package-lock.json > packages > < > > dependencies
1 {
2   "name": "frontend",
3   "version": "0.0.0",
4   "lockfileVersion": 3,
5   "requires": true,
6   "packages": {
7     "": {
8       "name": "frontend",
9       "version": "0.0.0",
10      "dependencies": [
11        "tailwindcss@vite": "4.1.8",
12        "lucide-react": "0.514.0",
13        "react": "19.1.0",
14        "react-dom": "19.1.0",
15        "tailwindcss": "4.1.8"
16      ],
17      "devDependencies": [
18        "eslint@^5.0.0",
19        "typescript@^4.1.2",
20        "vite@^2.8.4",
21        "vitejs/plugin-react": "4.4.1",
22        "eslint": "7.25.0",
23        "eslint-plugin-react-hooks": "5.2.0",
24        "eslint-plugin-react-refresh": "0.4.19",
25        "globals": "16.0.0",
26        "vite": "3.5.5"
27      ],
28      "node_modules/@amp-project/remapping": {
29        "version": "2.3.0",
30        "resolved": "https://registry.npmjs.org/@amp-project/remapping/-/remapping-2.3.0.tgz",
31        "integrity": "sha512-3o1ZAPg+LTYoelqv0s5j02j0Ysd5u6npkV0M3x0t9ah73erkgYAmzu43xVfqcnLx69Kpg385LcAYw==",
32        "license": "Apache-2.0",
33        "dependencies": [
34          "@jridgewell/gen-mapping": "0.3.5",
35          "@jridgewell/trace-mapping": "0.3.24"
36        ]
37      }
38    }
39  }
40  "node_modules/@amp-project/remapping": {
41    "version": "2.3.0",
42    "resolved": "https://registry.npmjs.org/@amp-project/remapping/-/remapping-2.3.0.tgz",
43    "integrity": "sha512-3o1ZAPg+LTYoelqv0s5j02j0Ysd5u6npkV0M3x0t9ah73erkgYAmzu43xVfqcnLx69Kpg385LcAYw==",
44    "license": "Apache-2.0",
45    "dependencies": [
46      "@jridgewell/gen-mapping": "0.3.5",
47      "@jridgewell/trace-mapping": "0.3.24"
48    ]
49  }

```

Live Share Fetching data for better TypeScript Intellisense

File Explorer

- project1
- backend
- frontend
- node_modules
- public
- src
- assets
- components
- AdminPanel.jsx
- ChatInterface.jsx
- TypingIndicator.jsx
- contexts
- App.css
- App.jsx
- index.css
- main.jsx
- .gitignore
- .eslintrc.json
- .index.html
- .package-lock.json
- .package.json
- README.md
- vite.config.js
- jp.resume.pdf
- old.py
- resume_praeven.pdf
- RohitResume.pdf
- project2
- project3
- OUTLINE
- TIMELINE

Line 14, Col 32 Spaces: 2 UTF-8 LF JSON

ENG IN 11:47 07-07-2025

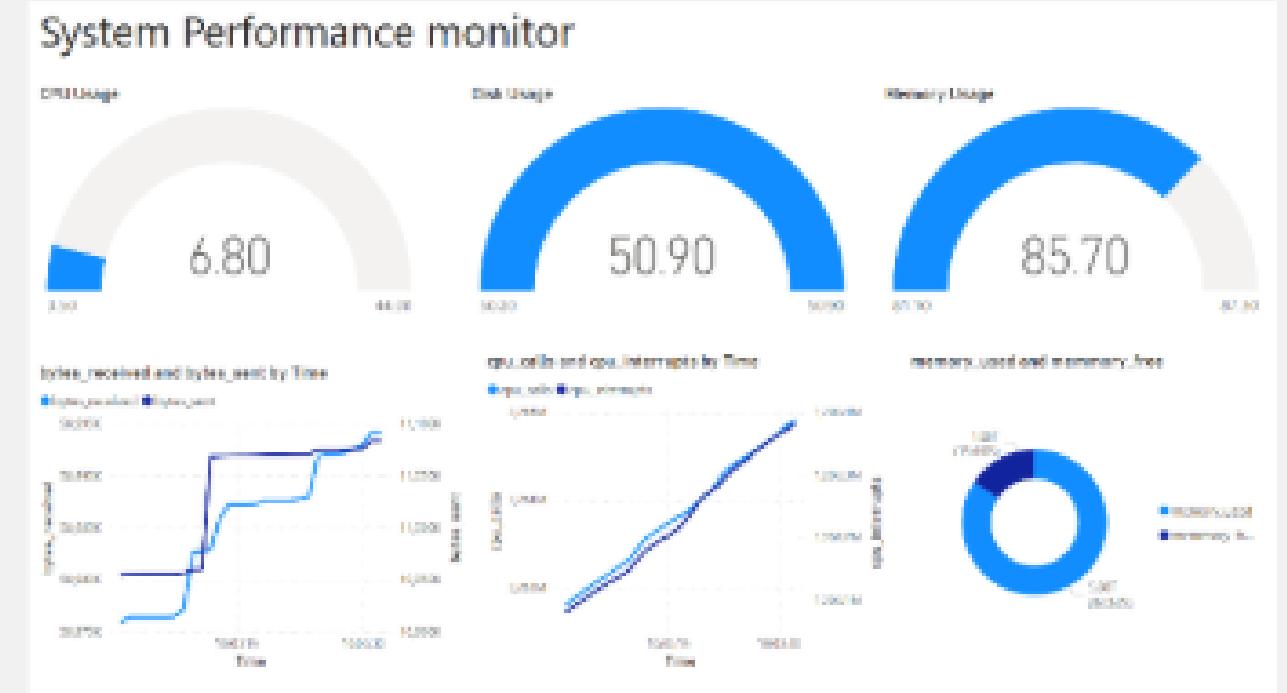
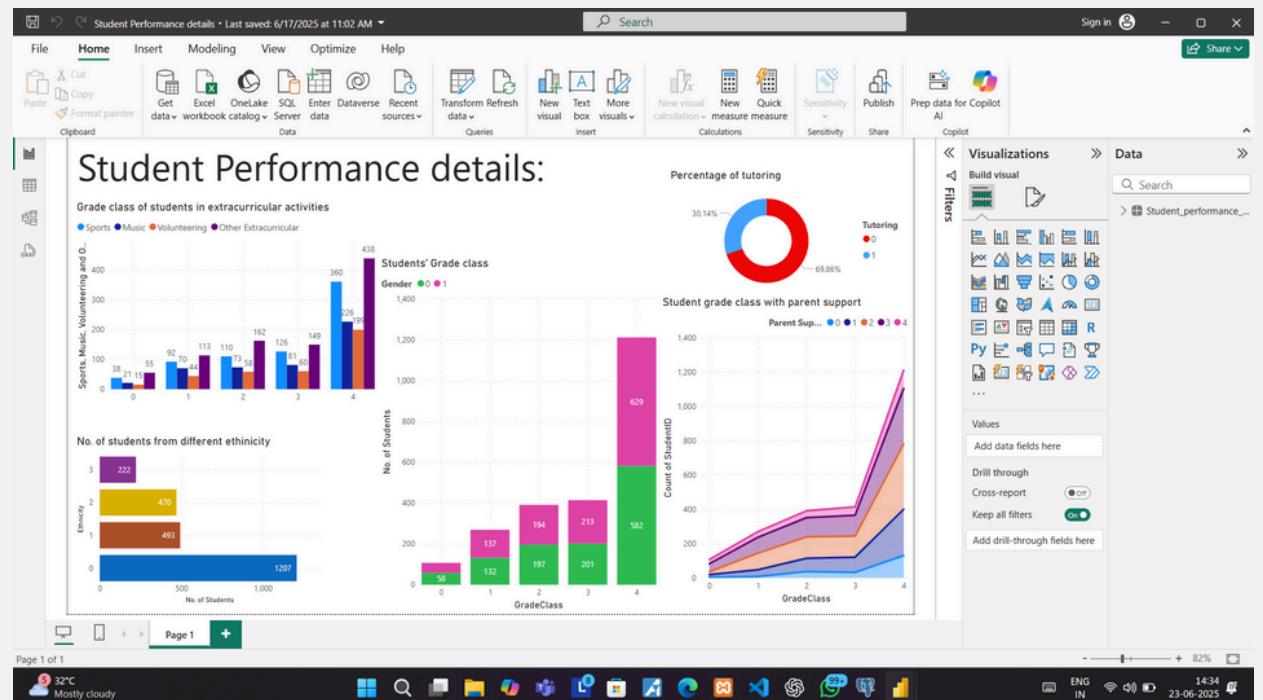


Project 2

System Performance Monitoring with Power BI

– Real-Time Analytics & Storage

This project involves building a real-time system monitoring solution using Power BI that detects and analyzes a computer's performance metrics such as CPU usage, memory consumption, disk activity, and network usage. A background script (written in Python or PowerShell) continuously collects system data and stores it in a PostgreSQL or SQL Server database. Power BI is then connected to this database to visualize the live data through interactive dashboards, enabling users to monitor performance trends and identify bottlenecks in real time. This setup provides both real-time insights and historical data analysis for improved system management.



```

# powerbipy.py
# D:\Dp_world\praveen>project2> powerbipy.py ...
1 import psutil
2 import time
3 import pyodbc
4 import socket
5 con = pyodbc.connect('Driver={SQL Server};'
6                      'Server=192.168.89.198\SQLEXPRESS;'
7                      'Database=System Information;'
8                      'Trusted_Connection=yes;')
9
10 cursor = con.cursor()
11
12 while True:
13     cpu_usage = psutil.cpu_percent()
14     memory_usage = psutil.virtual_memory()[2]
15     cpu_interrupts = psutil.cpu.stats()[1]
16     cpu_calls = psutil.cpu.stats()[3]
17     memory_used = psutil.virtual_memory()[3]
18     memory_free = psutil.virtual_memory()[4]
19     bytes_sent = psutil.net_io_counters()[0]
20     bytes_free = psutil.net_io_counters()[1]
21     disk_usage = psutil.disk_usage('/').percent
22     host_ip=socket.gethostname()
23
24     cursor.execute('insert into Performance values (GETDATE(),'
25                    + str(cpu_usage) +
26                    + str(memory_usage) +
27                    + str(cpu_interrupts) +
28                    + str(cpu_calls) +
29                    + str(memory_used) +
30                    + str(memory_free) +
31                    + str(bytes_sent) +
32                    + str(bytes_free) +
33                    + str(disk_usage) +
34                    + '""'+ host_ip +'")')
35
36     con.commit()
37     time.sleep(1)

```



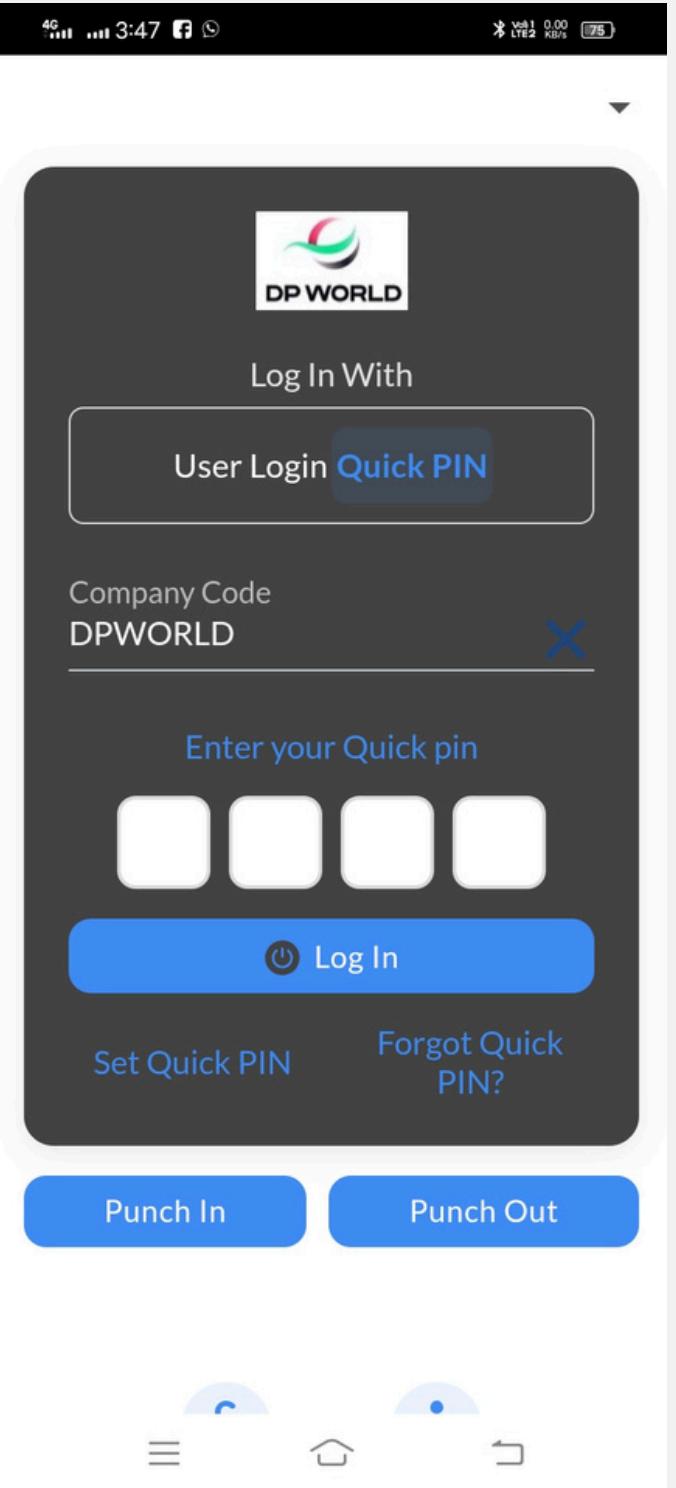
Reference Video



Project 3

Real-Time Attendance Tracking App with Geolocation – DP World Replica

This project is a React Native mobile application designed to replicate the DP World attendance system, featuring real-time geolocation tracking. When an employee logs their attendance, the app captures their current GPS location and timestamp, then securely stores this data in a backend database. The app ensures accurate and location-verified attendance records, offering features like employee role-based access, shift details, and punch-in/punch-out functionality. It provides a reliable and efficient solution for managing workforce attendance in real time.



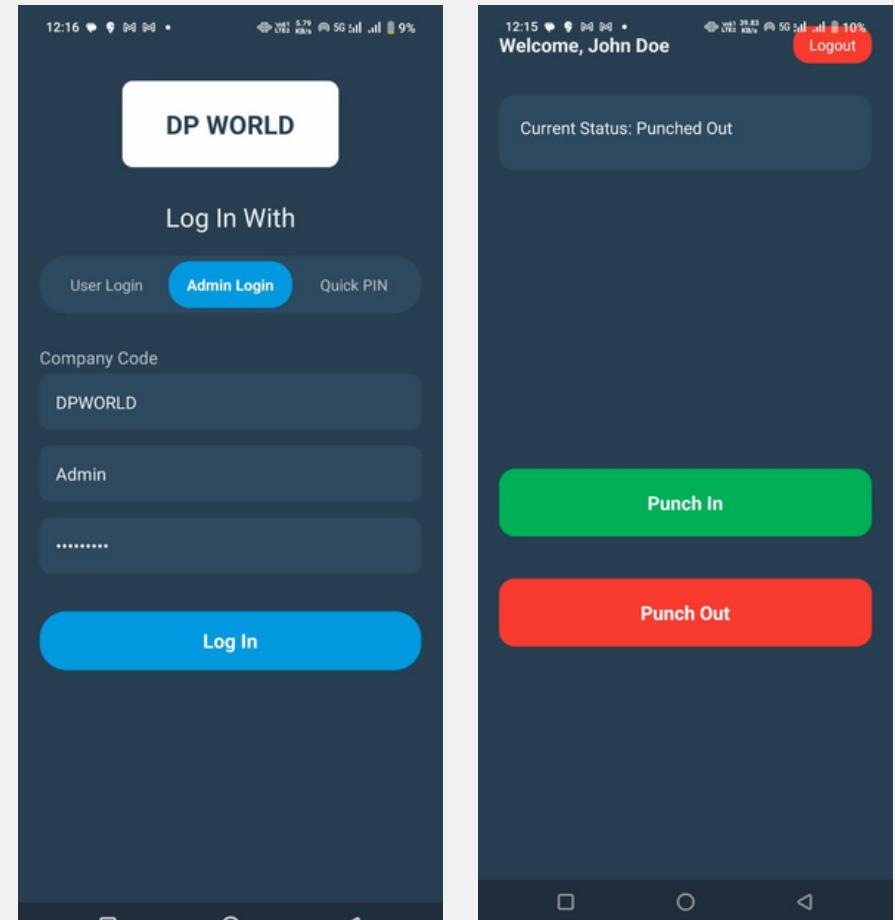
```
const App = () => {
  const [currentuser, setCurrentuser] = useState(null);
  const [userType, setUserType] = useState(null);

  useEffect(() => {
    checkAuthStatus();
  }, []);

  const handleLogin = (user, type) => {
    setCurrentuser(user);
    setUserType(type);
  };

  const handleLogout = () => {
    authservice.logout();
    setCurrentuser(null);
    setUserType(null);
  };

  const renderScreen = () => {
    if (currentuser) {
      return <LoginScreen onLogout={handleLogout} />;
    }
  };
}
```



```
import AsyncStorage from '@react-native-async-storage/async-storage';

export class AuthService {
  static async login(username, password, type = 'user') {
    try {
      const users = await this.getUsers();
      const user = users.find(
        u => u.username === username && u.password === password && u.type === type
      );

      if (user) {
        await AsyncStorage.setItem('currentUser', JSON.stringify(user));
        return user;
      }
      return null;
    } catch (error) {
      throw new Error('Login failed');
    }
  }

  static async loginWithPin(pin) {
    try {
      const users = await this.getUsers();
      const user = users.find(u => u.quickPin === pin);

      if (user) {
        await AsyncStorage.setItem('currentUser', JSON.stringify(user));
        return user;
      }
      return null;
    } catch (error) {
      throw new Error('PIN login failed');
    }
  }

  static async getCurrentUser() {
    try {
      const currentUserString = await AsyncStorage.getItem('currentUser');
      const currentUser = JSON.parse(currentUserString);
      return currentUser;
    } catch (error) {
      return null;
    }
  }
}
```

The pgAdmin 4 interface shows a connection to the "public.employee_attendance/Dp_world/postgres@PostgreSQL_17" database. The Object Explorer pane displays the schema structure, including tables like "dp_world_locations", "employee_attendance", "user_permissions", and "users". The Query pane contains the following SQL query:

```
SELECT * FROM public.employee_attendance
ORDER BY id ASC
```

The Data Output pane displays the results of the query:

	id [PK] integer	employee_id	employee_name	timestamp	action	selected_location	gps_coordinates	admin_own
1	1	1	John Doe	2024-06-19 13:30:00+05:30	punch_in	Main Office	25.2048, 55.2708	false
2	2	1	John Doe	2024-06-19 22:30:00+05:30	punch_out	Main Office	25.2048, 55.2708	false
3	3	1	John Doe	2025-06-20 12:40:53.109+05:30	punch_in	Main Office	13.101593423133739, 80.29688539675735	false
4	4	2	Admin User	2025-06-20 12:42:14.671+05:30	punch_in	Admin Office	13.101593423133739, 80.29688539675735	false
5	5	2	Admin User	2025-06-20 12:47:06.369+05:30	punch_in	Admin Office	37.4217937, -122.083922	false
6	6	2	Admin User	2025-06-20 12:51:09.667+05:30	punch_out	Admin Office	37.4217937, -122.083922	false
7	7	2	Admin User	2025-06-20 12:51:58.683+05:30	punch_out	Admin Office	37.4217937, -122.083922	false
8	8	2	Admin User	2025-06-20 16:01:43.241+05:30	punch_in	Admin Office	13.10164904636692, 80.29687251982277	false



Project 4

PHP - Project

This project is a web application developed using PHP that delivers dynamic content to users based on predefined logic and user interactions. It features a clean and responsive interface, integrates with a backend database (like MySQL or PostgreSQL), and supports basic CRUD operations. The project demonstrates core PHP functionalities such as form handling, session management, server-side validation, and real-time data rendering, making it a solid example of a content-driven PHP-based web solution.



Project 5

Interactive Data Analytics Dashboard in R with Live Stock-Style Visualization

Here's a one-paragraph description for your R programming project: This project is built using R programming and allows users to upload CSV or Excel files to automatically generate analytics and visualizations. Upon file upload, the system processes the data using R's powerful data manipulation libraries like dplyr and visualization tools like ggplot2 or plotly. It provides real-time insights, including trends, summaries, and live stock-style visual charts, helping users make quick and informed decisions based on their data. This project showcases R's capabilities in interactive data analysis and dashboard generation.

D:/r project - Shiny
http://127.0.0.1:5133 Open in Browser

DataViz Pro -

Upload Data Data Preview Live Stock Data Visualizations Analytics Insights

Upload Excel File

Drop your Excel file here
Supports .xlsx and .xls files

Choose File AAL.xlsx Upload complete

File uploaded successfully!
Proceed to Data Preview

Error reading file: 'arg' should be one of "default", "message", "warning", "error"

© 2025 DataViz Pro Enhanced | Live Stock Data Integration | Contact: developer@datavizpro.com | Version 2.0

33°C Partly sunny ENG IN 12:44 07-07-2025

This screenshot shows the DataViz Pro application interface. On the left is a sidebar with navigation links: Upload Data, Data Preview, Live Stock Data, Visualizations, Analytics, and Insights. The main area is titled 'Upload Excel File' with a large dashed box for dropping files. Below it, a message says 'File uploaded successfully!' and a button 'Proceed to Data Preview'. A small error message box is visible at the bottom right. At the bottom, there's a footer with the copyright information and system status (33°C, Partly sunny, ENG IN, 12:44, 07-07-2025).

D:/r project - Shiny
http://127.0.0.1:5133 Open in Browser

DataViz Pro -

Object Explorer Query History Scratch Pad

public.employee_attendance/Dp_world/postgres/PostgreSQL 17 X

Query

```
1 SELECT * FROM public.employee_attendance
2 ORDER BY id ASC
```

Data Output Messages Notifications

id [PK] integer	employee_id text	employee_name text	timestamp timestamp with time zone	action text	selected_location text	gps_coordinates text	admin_low boolean
1	1	John Doe	2024-06-19 13:30:00+05:30	punch_in	Main Office	25.2048, 55.2708	false
2	1	John Doe	2024-06-19 22:30:00+05:30	punch_out	Main Office	25.2048, 55.2708	false
3	1	John Doe	2025-06-20 12:45:53.109+05:30	punch_in	Main Office	13.101593423133739, 80.29688539675735	false
4	2	Admin User	2025-06-20 12:47:14.671+05:30	punch_in	Admin Office	37.4217937,-122.083922	false
5	2	Admin User	2025-06-20 12:47:06.369+05:30	punch_in	Admin Office	37.4217937,-122.083922	false
6	2	Admin User	2025-06-20 12:51:09.667+05:30	punch_out	Admin Office	37.4217937,-122.083922	false
7	2	Admin User	2025-06-20 12:51:58.683+05:30	punch_out	Admin Office	37.4217937,-122.083922	false
8	2	Admin User	2025-06-20 16:01:43.241+05:30	punch_in	Admin Office	13.10164904636692, 80.29687251982277	false

Total rows: 22 Query complete 00:00:00.120 CRLF Ln 1, Col 1

High UV Now

ENG IN 12:18 07-07-2025

This screenshot shows the pgAdmin 4 interface. It has an 'Object Explorer' on the left with a tree view of database objects. The central pane displays a query result table for 'public.employee_attendance'. The table has columns: id, employee_id, employee_name, timestamp, action, selected_location, gps_coordinates, and admin_low. The results show several entries of punches in and out for employees with IDs 1 and 2 across different dates and locations.

D:/r project - Shiny
http://127.0.0.1:5133 Open in Browser

DataViz Pro -

Upload Data Data Preview Live Stock Data Visualizations Analytics Insights

Stock Price Chart

\$159.09 Current Price - DEMO -1.63 Daily Change (-1.02%) 4.3M Volume Traded

Stock Price Chart - DEMO

This screenshot shows the DataViz Pro interface with a stock price chart. The top bar displays current price (\$159.09), daily change (-1.63 or -1.02%), and volume traded (4.3M). Below this is a candlestick chart titled 'Stock Price Chart - DEMO' showing price over time from April 6 to June 29. The chart includes moving average lines (MA 5 and MA 20). The bottom section is labeled 'Live Stock Data'.

D:/r project - Shiny
http://127.0.0.1:5133 Open in Browser

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

app.R

```
1 # D:/r project - Excel Analytics Dashboard with Live Stock Data
2 # Enhanced R Shiny application for Excel data visualization and live stock market data
3
4 # Load required libraries
5 library(shiny)
6 library(shinydashboard)
7 library(DT)
8 library(plotly)
9 library(readxl)
10 library(dplyr)
11 library(lubridate)
12 library(corrplot)
13 library(VIM)
14 library(shinyCSSloaders)
15 library(shinyWidgets)
16 library(httr)
17 library(jsonlite)
18 library(lubridate)
19 library(reactive)
20
21 # Helper function to calculate mode
22 mode <- function(x) {
23   ux <- unique(x)
24   ux[which.max(tabulate(match(x, ux)))]
25 }
```

Environment is empty

Files Plots Packages Help Viewer Presentation

Home Book1.xlsx Custom Office Templates Dash Default.rdp desktop.ini iISExpress My Web Sites Praveen Talent hub.xlsx praveen.pbx Shortcut to Documents (OneDrive - Personal).lnk SQL Server Management Studio SQL Server 2017 Visual Studio 2017 Visual Studio 2022 Preview WindowsPowerShell

Console Terminal Background Jobs

```
R 4.5.1 - /-
High 0
Low 0
Close 0
Adj Close 0
Volume 0
```

warning: Error in gather: could not find function "gather"

```
103: arrange
102: %
101: renderText [D:/r project/app.R#929]
100:
87: renderFunc
86: output$insights_content
1: shiny::runApp
```

ENG IN 12:44 07-07-2025

This screenshot shows the RStudio interface. The left pane shows an R script named 'app.R' with code for an R Shiny application. The right pane shows the 'Environment' tab which is currently empty. The bottom pane shows the 'Console' and 'Terminal' tabs.

Project 6



HR and Candidate Feedback Management System using ASP.NET

This project is a web-based HR and Candidate Feedback System developed using ASP.NET, designed to streamline the interview evaluation process. It allows HR personnel and interviewers to log in securely, view scheduled interviews, and submit structured feedback for each candidate. The system includes features such as auto-filled candidate details, role-based access, star rating components, and comment sections to capture both qualitative and quantitative feedback. All feedback data is stored in a backend SQL database, enabling easy retrieval and reporting. This project enhances transparency, consistency, and efficiency in managing interview outcomes.

Visual Studio IDE showing the InterviewFeedbackSystem project structure. The code in Program.cs configures the application to use PostgreSQL as the database.

```
project > dotnet > InterviewFeedbackSystem > Programs > Program.cs
1 using InterviewFeedbackSystem.Data;
2 using InterviewFeedbackSystem.Models;
3 using Microsoft.AspNetCore.Identity;
4 using Microsoft.EntityFrameworkCore;
5 using Npgsql.EntityFrameworkCore.PostgreSQL;
6
7 var builder = WebApplication.CreateBuilder(args);
8
9 // Add services to the container.
10 var connectionString = builder.Configuration.GetConnectionString("DefaultConnection") ??
11     "host=localhost;port=5000;database=interviewfeedback;username=praveen;password=praveen@200d";
12
13 builder.Services.AddDbContext<ApplicationDbContext>(options =>
14     options.UseNpgsql(connectionString)); // changed to PostgreSQL
15
16 builder.Services.AddDatabaseDeveloperPageExceptionFilter();
17
18 builder.Services.AddDefaultIdentity<ApplicationUser>(options =>
19     options.SignIn.RequireConfirmedAccount = false;
20     options.Password.RequiredLength = 6;
21     options.Password.RequireLowercase = false;
22     options.Password.RequireUppercase = false;
23     options.Password.RequireNonAlphanumeric = false;
24
25     options.Password.RequiredOrLowercase = false;
26 );
27
28 .AddRoles<IdentityRole>()
29 .AddInMemoryStores<ApplicationDbContext>();
30
31 builder.Services.AddControllersWithViews();
32
33 var app = builder.Build();
34
35 // Configure the HTTP request pipeline.
36 if (app.Environment.IsDevelopment())
37 {
```

Visual Studio IDE showing the 20250703071922_InitialCreate.cs migration code. This migration creates two tables: AspNetUsers and AspNetRoles.

```
project > dotnet > InterviewFeedbackSystem > Migrations > 20250703071922_InitialCreate.cs
1 using System;
2 using Microsoft.EntityFrameworkCore.Migrations;
3 using Npgsql.EntityFrameworkCore.PostgreSQL.Metadata;
4
5 #nullable disable
6
7 namespace InterviewFeedbackSystem.Migrations
8 {
9     // inherit doc
10     public partial class InitialCreate : Migration
11     {
12         protected override void Up(MigrationBuilder migrationBuilder)
13         {
14             migrationBuilder.CreateTable(
15                 name: "AspNetRoles",
16                 columns: table => new
17                 {
18                     Id = table.Column<string>(type: "text", nullable: false),
19                     Name = table.Column<string>(type: "character varying(256)", maxLength: 256, nullable: true),
20                     NormalizedName = table.Column<string>(type: "character varying(256)", maxLength: 256, nullable: true),
21                     ConcurrencyStamp = table.Column<string>(type: "text", nullable: true)
22                 },
23                 constraints: table =>
24                 {
25                     table.PrimaryKey("PK_AspNetRoles", x => x.Id);
26                 });
27
28             migrationBuilder.CreateTable(
29                 name: "AspNetUsers",
30                 columns: table => new
31                 {
32                     Id = table.Column<string>(type: "text", nullable: false),
33                     FirstName = table.Column<string>(type: "text", nullable: true),
34                     LastName = table.Column<string>(type: "text", nullable: true),
35                     Department = table.Column<string>(type: "text", nullable: true),
36                     CreatedAt = table.Column<DateTime>(type: "timestamp with time zone", nullable: true)
37                 },
38                 constraints: table =>
39                 {
40                     table.PrimaryKey("PK_AspNetUsers", x => x.Id);
41                 });
42         }
43     }
44 }
```

Interview Feedback System HR Dashboard. It displays the total feedback count (5), average rating (3.0), and recommendation rate (40%). It also shows recent feedback from candidates praveen and Robert Wilson.

Candidate	Position	Rating	Date	Actions
praveen	XZ	★☆☆☆☆	2.0	Jul 04, 2025
praveen	XZ	★☆☆☆☆	1.0	Jul 04, 2025
Robert Wilson	Data Analyst	★★★★☆	3.0	Jul 02, 2025

Interview Feedback System Submit Interview Feedback page. It allows users to enter candidate information, interview details, and rate their experience.

Position Applied	Interviewer Name
Software Developer	John Doe

pgAdmin 4 interface showing the public schema of the InterviewFeedbackDB. It lists tables like AspNetUsers, InterviewFeedbacks, and Logins, along with their columns and data.

Column	Type	Comments
Id	text	PK
FirstName	text	
LastName	text	
Department	text	
CreatedAt	timestamp with time zone	
UserName	character varying(256)	
NormalizedUserName	character varying(256)	

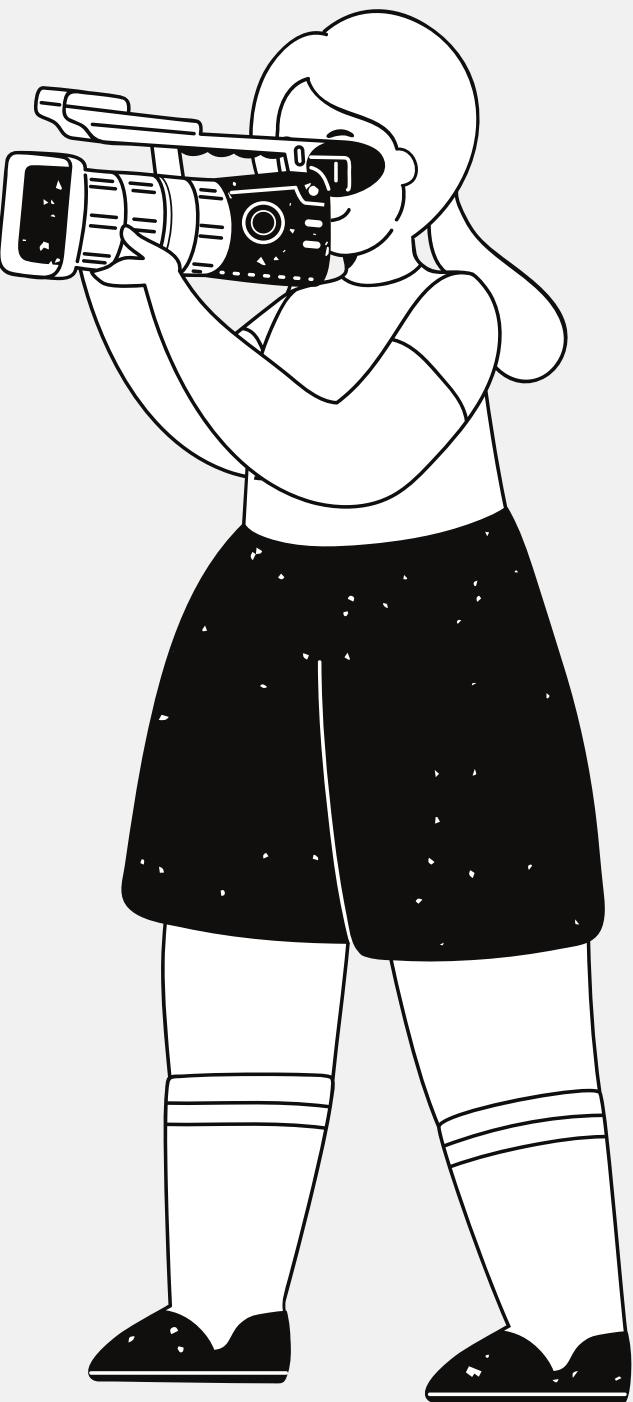
Server



As part of my server-side and networking fundamentals training, I gained hands-on knowledge of key network infrastructure components including core switches, distribution switches, and routers. I learned how these devices manage data flow within enterprise networks, enable efficient routing, and maintain network reliability. Additionally, I explored fiber optic technology, understanding its role in high-speed data transmission and backbone connectivity in data centers. This foundational knowledge strengthened my understanding of how server-side systems communicate securely and efficiently across distributed networks.

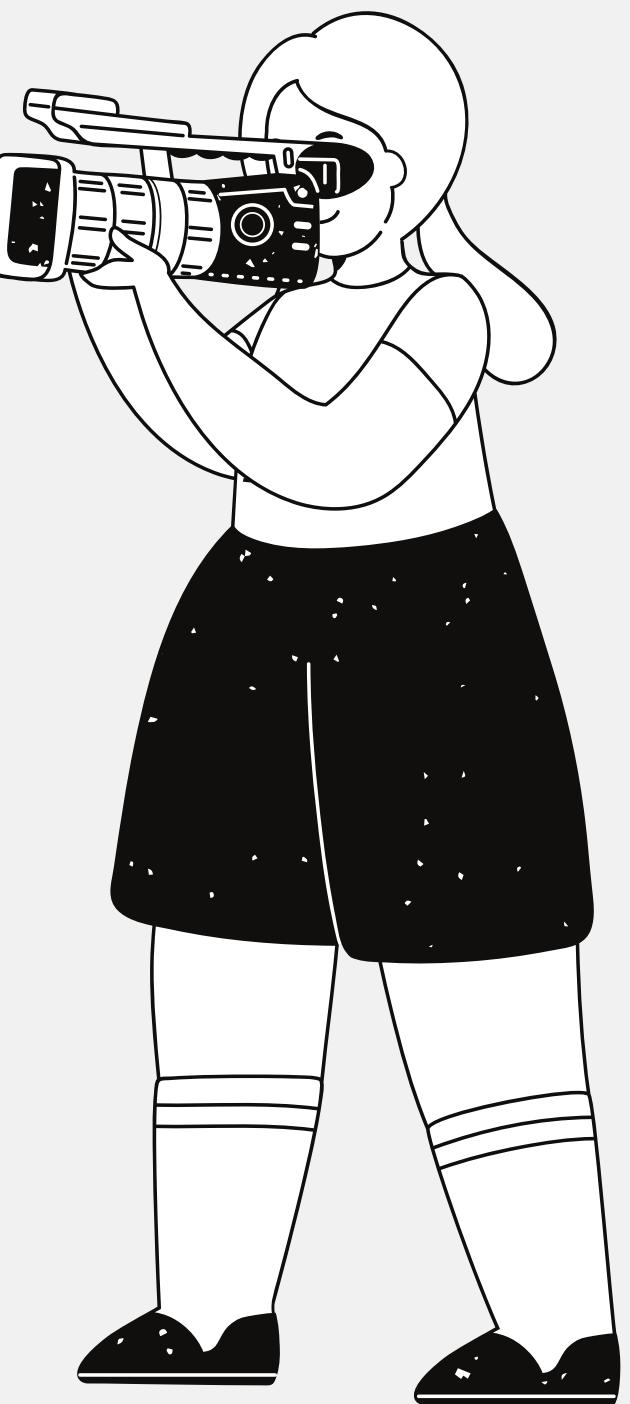
System Administrator

Experienced in managing network infrastructure, server-side operations, and system security as a System Administrator.



As a System Administrator, I developed a strong understanding of administrative responsibilities such as user management, system configuration, and policy enforcement. I also provided technical support by troubleshooting hardware, software, and network issues, ensuring minimal downtime. Additionally, I offered desktop support to end-users by resolving system errors, configuring workstations, and maintaining a stable and secure computing environment across the organization.

Networking



During my time at DP World, I gained valuable insights into how enterprise networking operates within a large-scale organization. I learned about the structured network architecture involving core switches, distribution switches, and routers, and how they are configured to manage internal communication and external connectivity efficiently. I also understood the role of VLANs, IP addressing, and firewall configurations in ensuring secure and segmented network access. Additionally, I explored how fiber optic connections are used for high-speed data transfer between critical infrastructure components, contributing to reliable and scalable network performance.

Additional Information



CNAPP (Cloud-Native Application Protection Platform)

CNAPP is an integrated security solution designed to protect cloud-native applications across their entire lifecycle. It combines multiple tools such as container security, workload protection, and cloud security posture management (CSPM), ensuring applications are secure from development to production.

ZT (Zero Trust Architecture)

Zero Trust is a modern cybersecurity framework that enforces strict identity verification for every user and device, whether inside or outside the network. It operates on the principle of "never trust, always verify," minimizing the risk of unauthorized access and lateral movement within the system.

Flash Database

Flash databases utilize solid-state drives (SSDs) or NVMe storage to deliver extremely fast data access speeds. They are ideal for workloads that require real-time processing, such as high-frequency trading, IoT data streams, and instant analytics.

SAN (Storage Area Network)

A SAN is a high-speed, specialized network that provides block-level storage access to servers. It separates storage resources from the local network, enabling efficient, scalable, and centralized storage management in enterprise environments.

SASE (Secure Access Service Edge)

SASE is a cloud-delivered architecture that merges network security services like secure web gateways, firewall-as-a-service, and Zero Trust Network Access (ZTNA) with wide area networking (WAN). It allows secure and efficient remote access to applications and data from any location.

Streaming GPU

Streaming GPU technology enables remote access to powerful graphical and AI processing capabilities hosted on cloud servers. This approach is used in gaming, machine learning, and visualization-heavy applications, allowing high-performance output without requiring local GPU hardware.

Data Lakehouse

A data lakehouse blends the scalability of data lakes with the performance and structure of data warehouses. It supports a variety of data formats and analytics types, enabling both real-time and batch processing for structured, semi-structured, and unstructured data.

Switch Configuration Modes

- Enable Mode: Grants elevated access privileges to view detailed device configurations and monitor operations.
- Configuration Mode: Allows administrators to modify system-level settings such as VLANs, routing protocols, interface behavior, and access control lists (ACLs).

Zodiac, ZingHR, Nagios

- **Zodiac**: A terminal operations software used to plan, track, and manage container handling and yard operations in logistics hubs like ports.
- **ZingHR**: A cloud-based Human Resource Management System (HRMS) used for handling payroll, attendance, recruitment, and workforce analytics.
- **Nagios** is a popular monitoring tool, and in DP World, a licensed version is used. However, for modern server monitoring, open-source tools like Grafana with Prometheus offer better dashboards, easier integration, and real-time metrics, making them more suitable for today's infrastructure needs.

Finance, NTT, Tower Control

- **Finance:** Learned about invoicing, budgeting, cost tracking, and the integration of financial workflows with operational systems for transparent reporting.
- **NTT (Network Troubleshooting Tools):** Hands-on experience with tools used to diagnose latency, packet loss, and routing issues in enterprise networks.
- **Tower Control:** A centralized monitoring and control center in logistics, managing container movements, crane operations, and overall terminal flow.

RTG (Rubber-Tyred Gantry Crane)

- RTG: Mobile cranes used in container yards for stacking and unstacking containers, offering flexibility in yard management.

QC (Quay Crane)

- QC: Large, fixed cranes located at the quay used to load and unload shipping containers from vessels, essential for port operations.

Roaster Team – Manual Manpower Allocation

- Observed how the roaster team manually plans and allocates workforce shifts based on terminal schedules, crane assignments, and workload distribution to ensure operational efficiency.

**Thank you
very much!**