

# Full Stack Development Tools Overview

► **Subtitle:** Postman | Eclipse | Java | HTML | CSS  
| GitHub | MySQL

**Presented by:** TEJASHWINI B M

**Institute:** CEC

**Date:** 08-10-2025

# Agenda

- ▶ Introduction
- ▶ Postman
- ▶ Eclipse
- ▶ Java
- ▶ HTML
- ▶ CSS
- ▶ GitHub
- ▶ MySQL
- ▶ Project Explanation
- ▶ Summary
- ▶ Q&A

# Introduction

## Overview of Full-Stack Development

- ▶ Full-stack development involves building both frontend (user interface) and backend (server, database, and logic) parts of a web application.
- ▶ A full-stack developer handles everything — from designing how a website looks to how it stores and processes data.

## Why Each Tool is Essential

- ▶ Frontend Tools (HTML, CSS, JavaScript, React, etc.) → Create responsive, user friendly interfaces.
- ▶ Backend Tools (Node.js, Express.js, etc.) → Handle logic, authentication, and data flow.
- ▶ Database (MongoDB, MySQL, etc.) → Store and retrieve data efficiently.
- ▶ Version Control (Git, GitHub) → Manage code and collaborate easily

## Tech Stack Covered in This Session

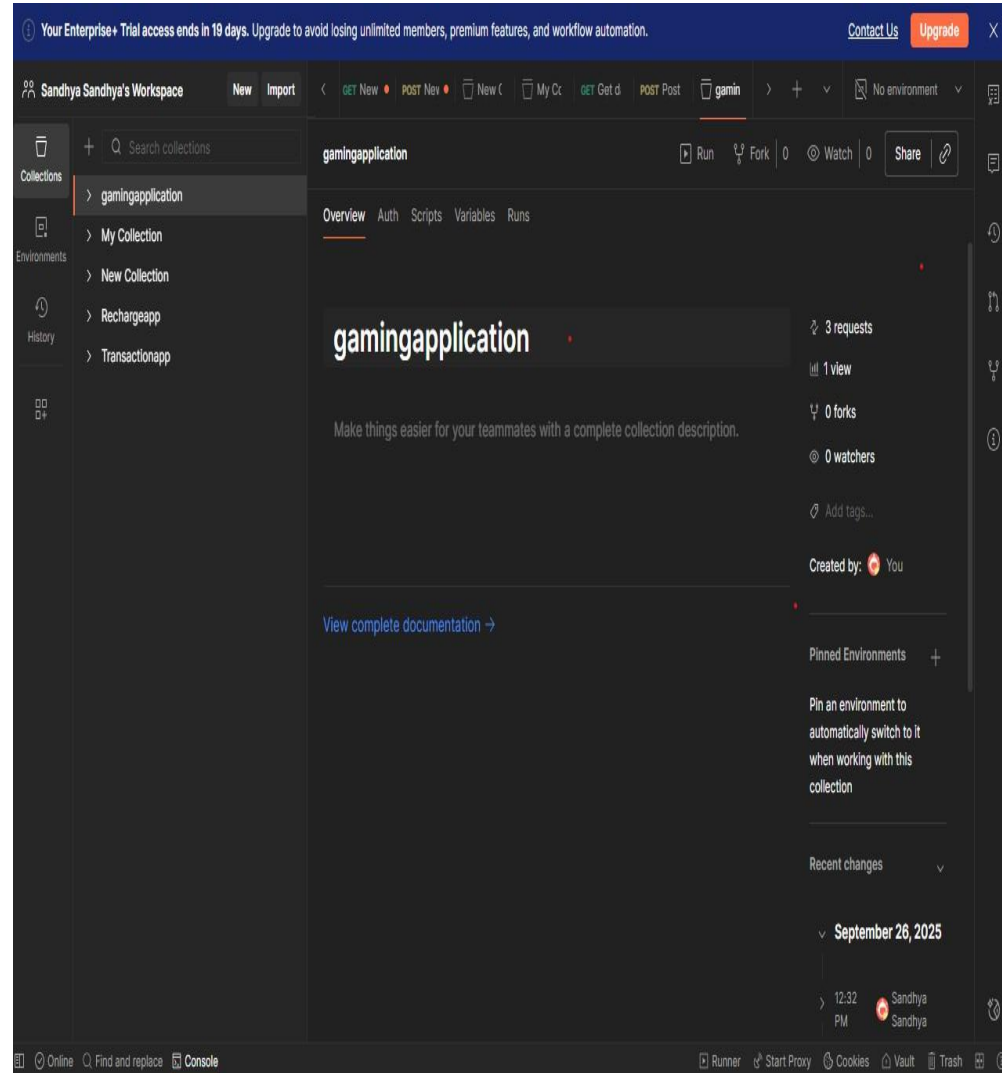
- ▶ Frontend: HTML, CSS, JavaScript, React
- ▶ Backend: Node.js, Express.js
- ▶ Database: MongoDB
- ▶ Tools: Git, GitHub, VS Code

# Postman

**Purpose:** API testing and debugging

## Key Features:

- ❖ Create and test REST API– Send requests (GET, POST, PUT, DELETE) and view responses.
- ❖ Automate test cases– Run pre-defined tests to check API performance and correctness.
- ❖ Use collections and environments– Organize APIs and manage variables for different setups (dev, test, prod)
- ❖ **Example:** Testing login API endpoint before UI integration → Send a POST request with username and password to check if login works properly before connecting it to the UI.



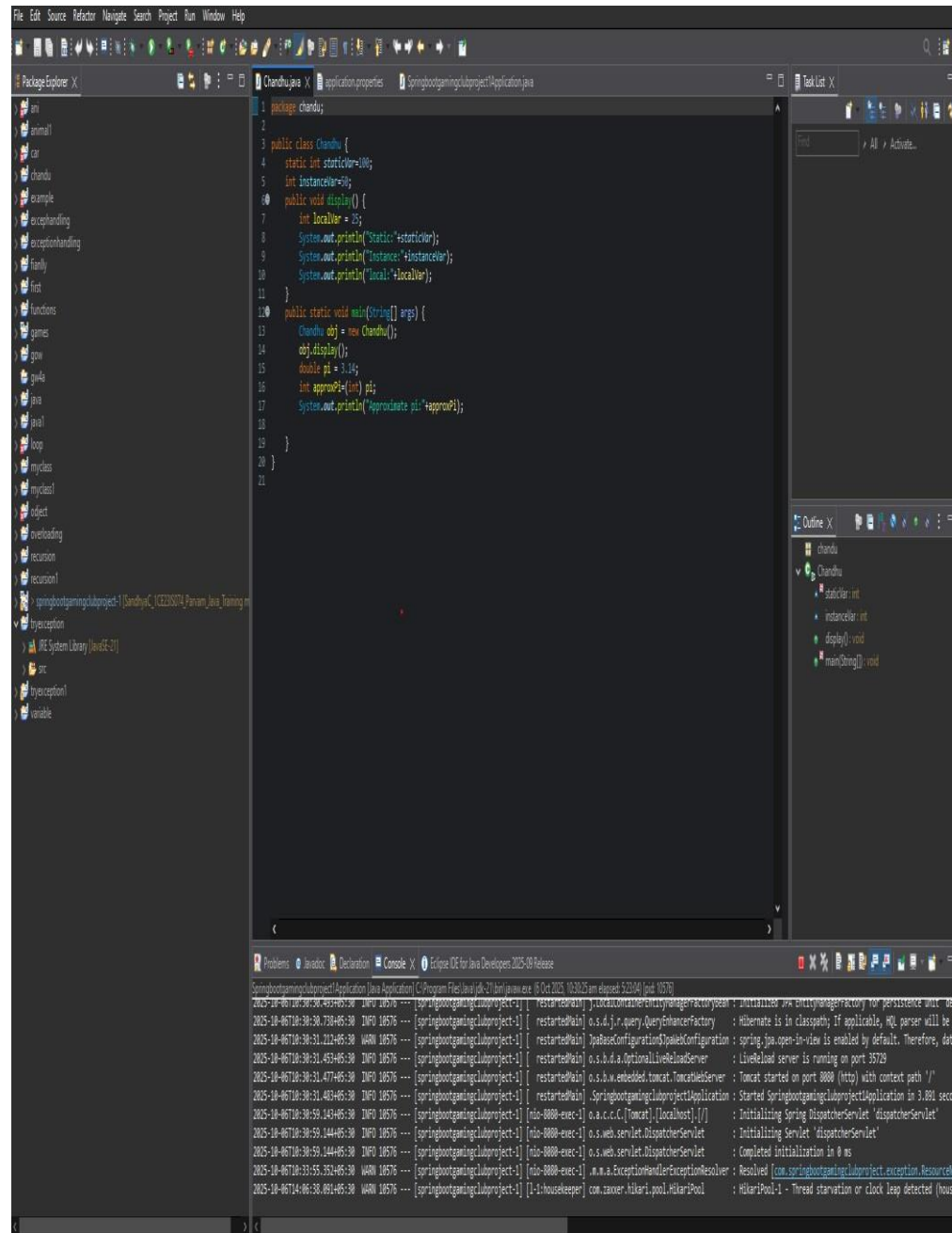
# Eclipse IDE

## Purpose

- A powerful development platform for Java and other programming languages.
- Commonly used for backend API development in enterprise applications.

## Key Features

- Code Editing, Debugging & Build Automation –Write, run, and debug Java projects efficiently.
- Supports Maven and Git Plugins –Manage dependencies and integrate version control seamlessly.
- Suitable for Backend API Development – Ideal for creating and testing server-side logic and RESTful services.



# JAVA

## Purpose:

- Core backend programming language

## Highlights:

- Object-Oriented (follows class and object concepts)
- Platform Independent (runs on any OS via JVM)
- Rich Library Support (extensive APIs for networking, I/O, etc.)
- Secure and Robust (automatic memory management)
- Multi-threaded (supports concurrent programming)
- Widely Used for Enterprise Applications

```
public class Main {  
    public static void  
    main(String[] args) {  
  
        System.out.println("Welcome to  
Full Stack Development!");  
    }  
}
```

# HTML

## Purpose:

- ❖ Provides the structure and layout of web pages

## Highlights:

- ❖ Stands for Hyper Text Mark up Language
- ❖ Defines headings, paragraphs, links, forms, and media
- ❖ Forms the skeleton of every website
- ❖ Works together with CSS (for design) and JS (for behavior)
- ❖ Uses tags and attributes to format content



The image shows a code editor with a file named `index.html` open. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13  <!--
14    manifest.json provides metadata used when your web app is installed on a
15    user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
16  -->
17  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
18  <!--
19    Notice the use of %PUBLIC_URL% in the tags above.
20    It will be replaced with the URL of the 'public' folder during the build.
21    Only files inside the 'public' folder can be referenced from the HTML.
22
23    Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
24    work correctly both with client-side routing and a non-root public URL.
25    Learn how to configure a non-root public URL by running 'npm run build'
26  -->
```

Below the code editor is a terminal window showing the output of a build process. It indicates that the application was compiled with warnings. The warning is from ESLint, specifically the `no-template-curly-in-string` rule, which was triggered in `src\api.js` at line 28:50. The warning message is: "Unexpected template string expression `no-template-curly-in-string`". The terminal also shows that Webpack compiled with 1 warning.



# CSS

## Purpose:

- Adds styling and design to HTML pages

## Highlights:

- Stands for Cascading Style Sheets
- Controls layout, colors, fonts, and spacing
- Separates content (HTML) from design (CSS)
- Supports responsive design for all screen sizes
- Enhances user interface and overall experience (UI/UX)



The image shows a code editor with a light theme. The main editor area displays an `index.html` file with the following content:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8" />
5   <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6   <meta name="viewport" content="width=device-width, initial-scale=1" />
7   <meta name="theme-color" content="#000000" />
8   <meta
9     name="description"
10    content="Web site created using create-react-app"
11  />
12  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13  <!--
14    manifest.json provides metadata used when your web app is installed on a
15    user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
16  -->
17  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
18  <!--
19    Notice the use of %PUBLIC_URL% in the tags above.
20    It will be replaced with the URL of the 'public' folder during the build.
21    Only files inside the 'public' folder can be referenced from the HTML.
22
23    Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
24    work correctly both with client-side routing and a non-root public URL.
25    Learn how to configure a non-root public URL by running 'npm run build'
26  -->
```

Below the code editor is a terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Compiled with warnings.

[eslint]
src\api.js
  Line 28:50: Unexpected template string expression  no-template-curly-in-string

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

WARNING in [eslint]
src\api.js
  Line 28:50: Unexpected template string expression  no-template-curly-in-string

webpack compiled with 1 warning
```

On the right side of the terminal window, there is a sidebar with icons for different command-line interfaces: `cmd`, `cmd`, `cmd`, and `node`. The `node` icon is currently selected.



# GITHUB

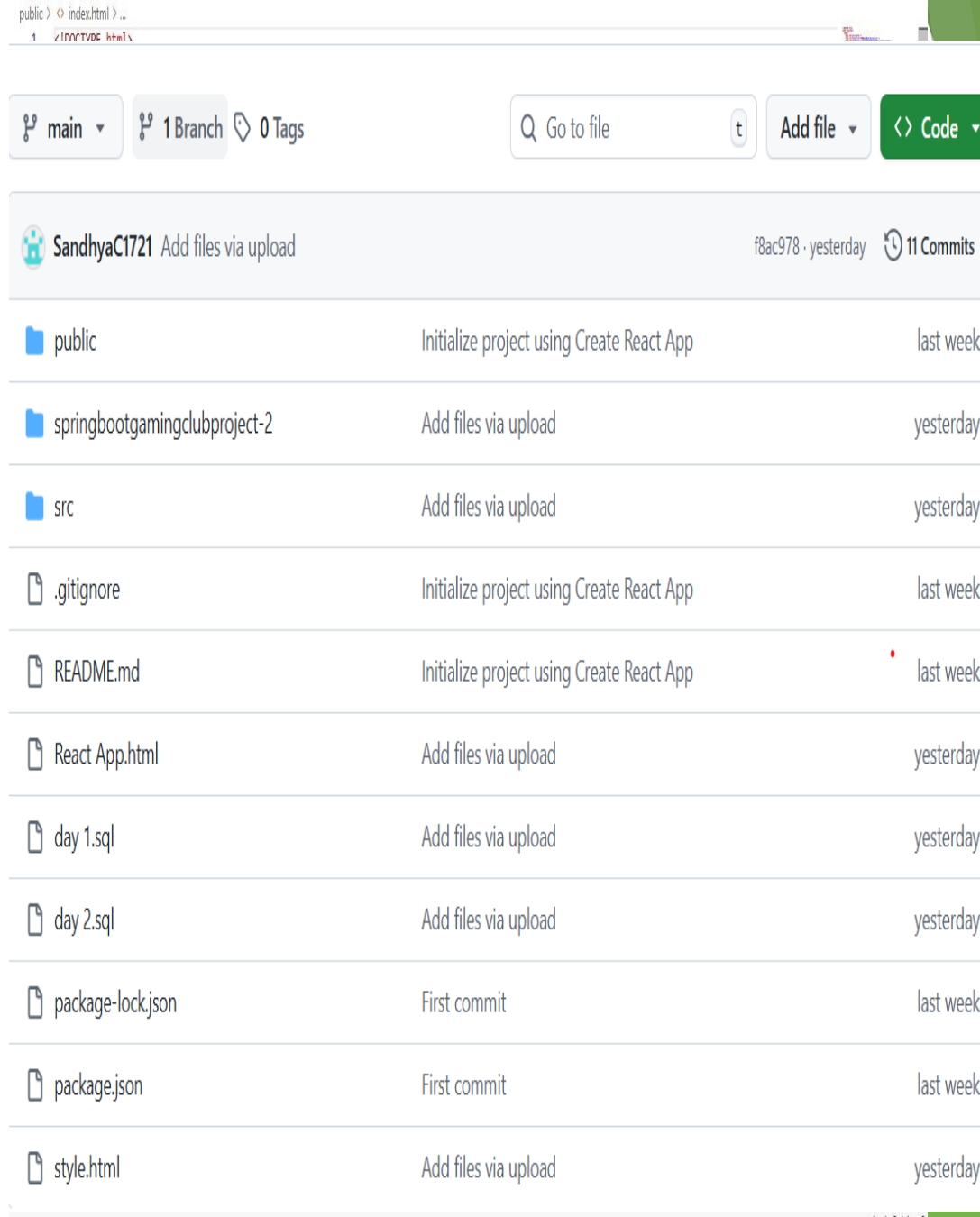
## Purpose:

Version control and team collaboration platform

## Features:

- Manage and track source code versions
- Supports branching and merging for team work
- Integrates with CI/CD pipelines for automation
- Enables pull requests and code reviews
- Provides GitHub Pages for hosting projects
- Cloud-based and accessible anywhere

**Example:** Store and share code projects  
Collaborate with developers through commits and branches



main	1 Branch	0 Tags	Go to file	Add file	Code
SandhyaC1721 Add files via upload			f8ac978 · yesterday	11 Commits	
public	Initialize project using Create React App	last week			
springbootgamingclubproject-2	Add files via upload	yesterday			
src	Add files via upload	yesterday			
.gitignore	Initialize project using Create React App	last week			
README.md	Initialize project using Create React App	last week			
React App.html	Add files via upload	yesterday			
day 1.sql	Add files via upload	yesterday			
day 2.sql	Add files via upload	yesterday			
package-lock.json	First commit	last week			
package.json	First commit	last week			
style.html	Add files via upload	yesterday			

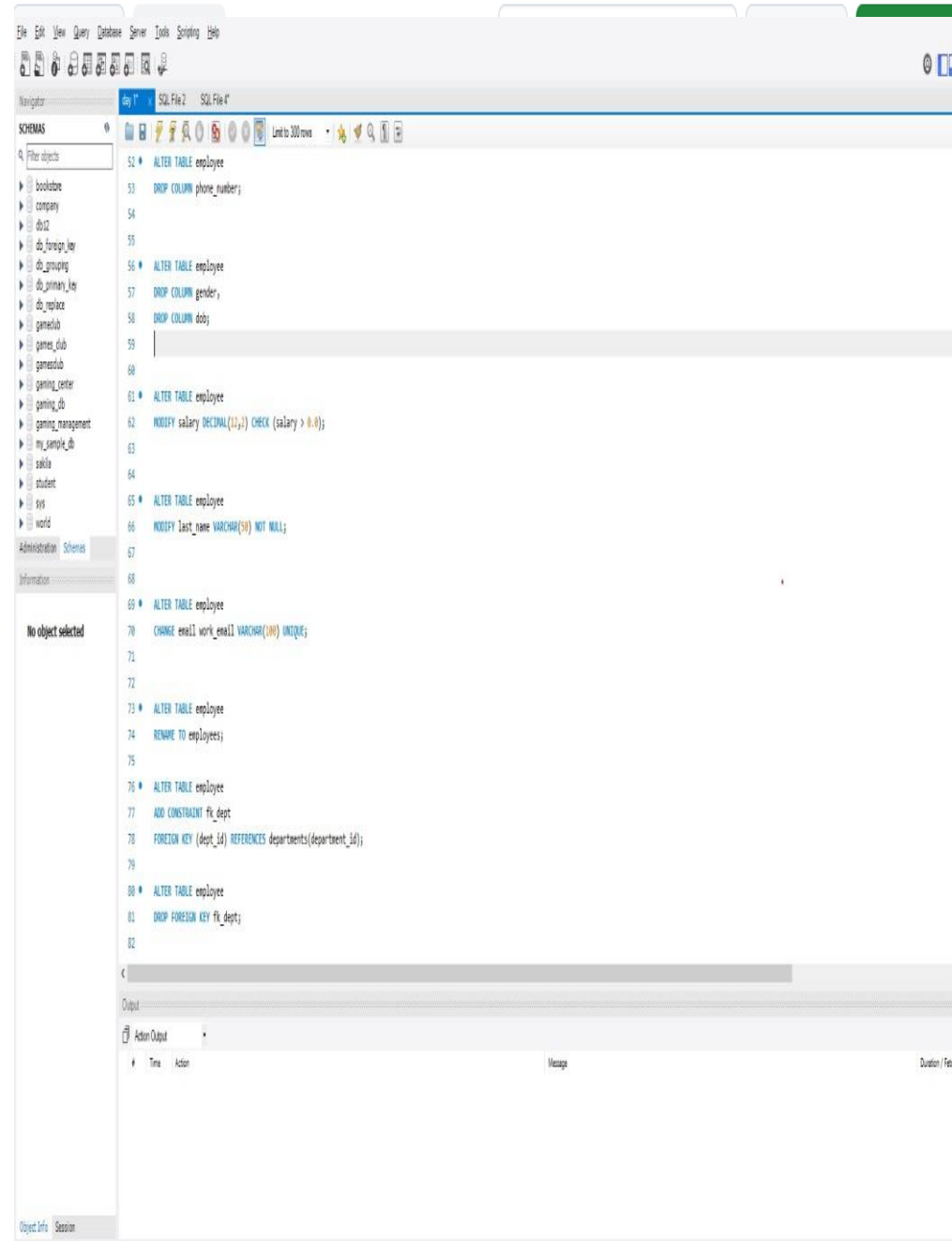
# MYSQL

## Purpose:

- Backend database used for storing and managing application data

## Highlights:

- Open-source Relational Database Management System (RDBMS)
- Uses SQL (Structured Query Language) for data operations
- Supports tables, relationships, and constraints
- Ensures data integrity and security
- Widely used with backend technologies like Java, PHP, and Python



# Slide 11: Project Explanation

**Name:** Employee Management System

**Overview:**

A web-based system to manage employee details, allowing CRUD operations through a Java backend and MySQL database

**.Tools Used and its Purpose:**

HTML/CSS

Frontend UI

Java (Eclipse)

Backend logic

MySQL

Data storage

Postman

API testing

GitHub

Version control

Eclipse IDE

Code development

**Workflow:**

**Frontend (HTML/CSS)** → Sends data to backend via form submission **.Backend (Java)** → Processes data and interacts with My SQL. **Database (MySQL)** → Stores and retrieves employee information **.Postman** → Used for testing backend **.APIs GitHub** → Used for source code management. **Outcome:** ✓ Complete CRUD-based application ✓ Integrated testing using Postman ✓ Database connectivity with MySQL ✓ Hosted and managed via GitHub

# SUMMARY

Tool

Role

Postman

API Testing

Eclipse

IDE for Development

Java

Backend Logic

HTML

Page Structure

CSS

Page Styling

GitHub

Version Control

MySQL

Database

# Q&A

QUESTIONS AND DISCUSSION

CONTACT:[[tejashwinibmtejashwinibm@gmail.com](mailto:tejashwinibmtejashwinibm@gmail.com)]