



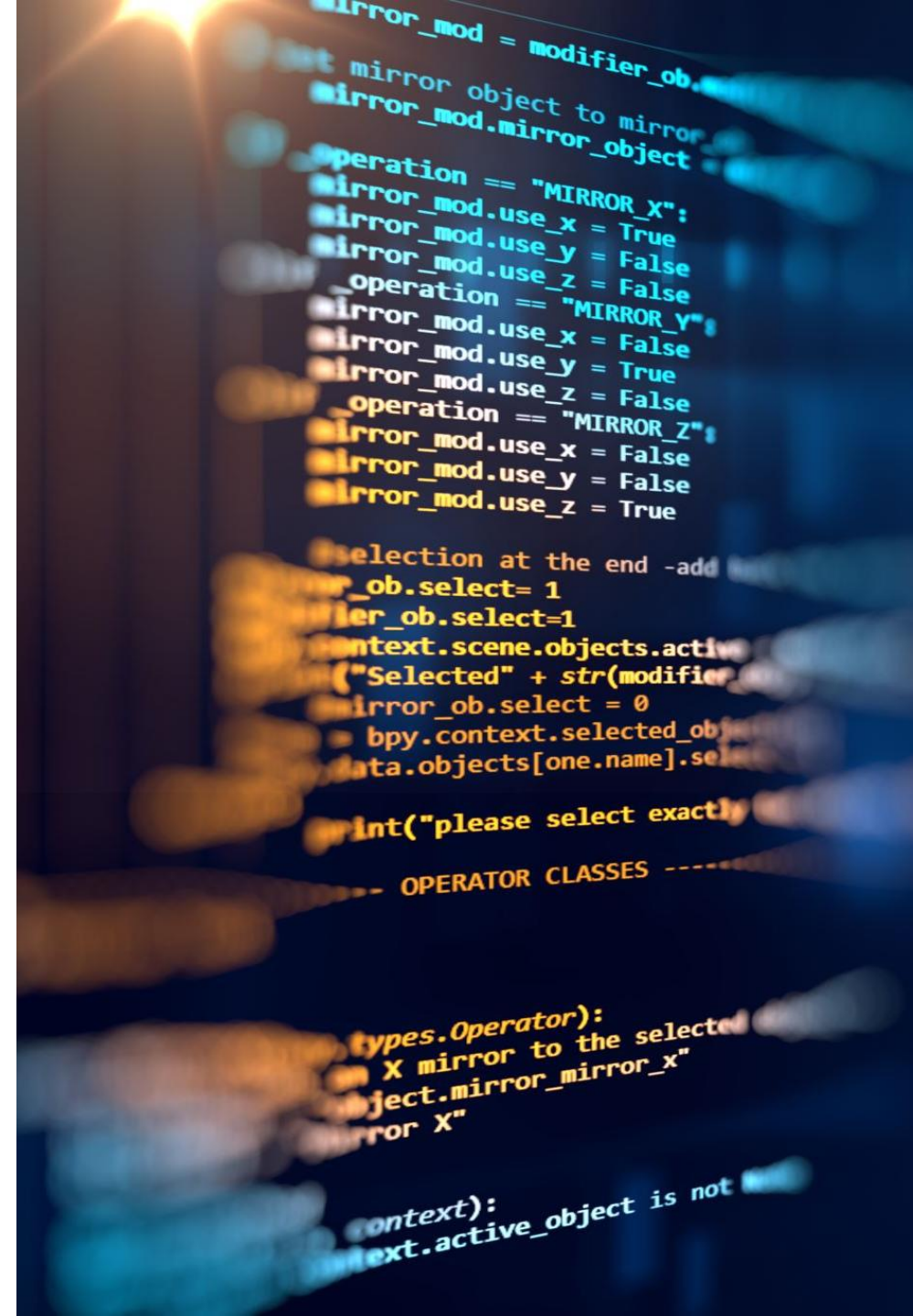
# SOFTWARE DEVELOPMENT AND DESIGN BASICS - II

BY KHAJA



# MOST COMMON SOFTWARE APPLICATION TYPES

- Operating Systems: The foundational software that manages hardware resources and provides a platform for other software to run (Microsoft Windows, macOS, Linux, Android, iOS)
- System Software: Software that facilitates the functioning of the computer system and supports application development (Device drivers, utilities, compilers, and interpreters)
- Application Software: We will be discussing this over the next slide



# APPLICATION SOFTWARE TYPES



**General-Purpose Applications:** Software used for common tasks and productivity. (Word processors, spreadsheets, web browsers)



**Specialized Applications:** Software designed for specific tasks or industries. (Graphic design software (e.g., Adobe Photoshop))



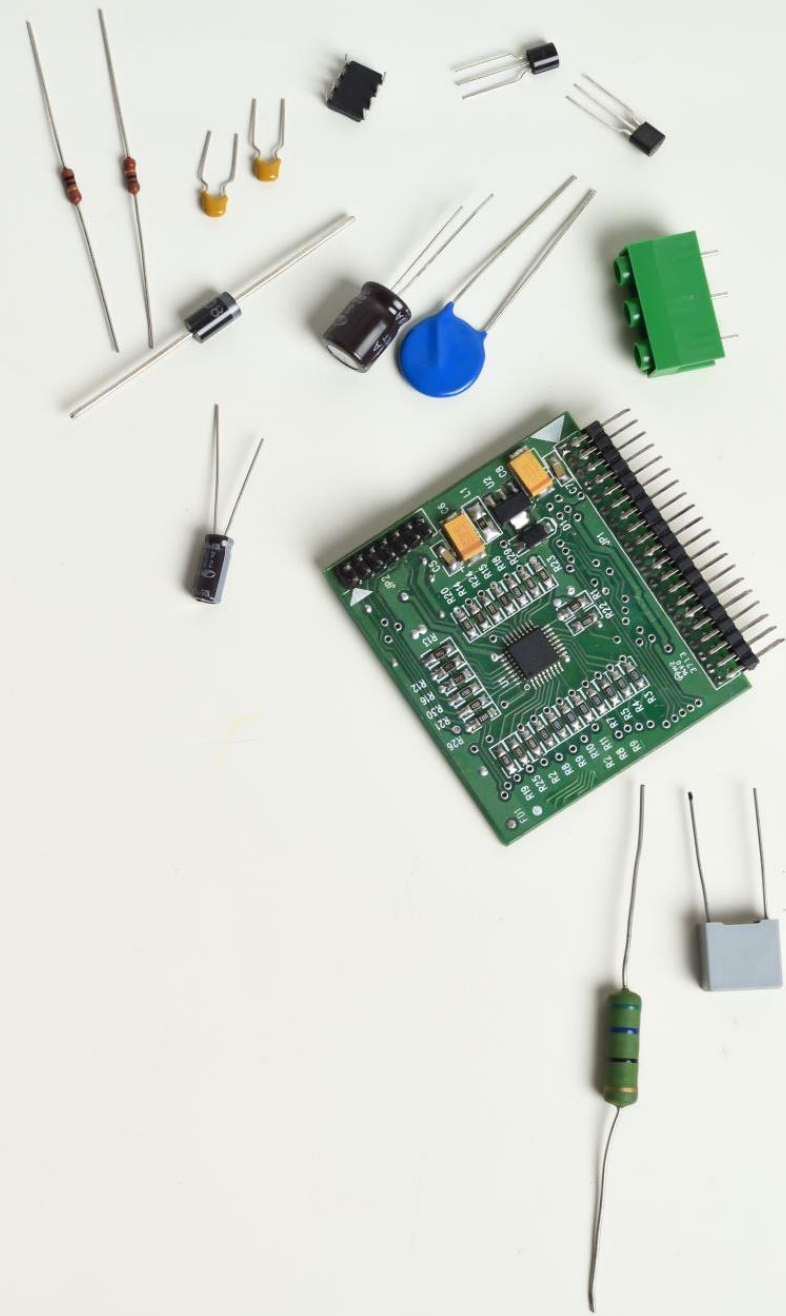
**Enterprise Software:** Software used by organizations to manage business operations. (Customer relationship management (CRM) software, enterprise resource planning (ERP) software)

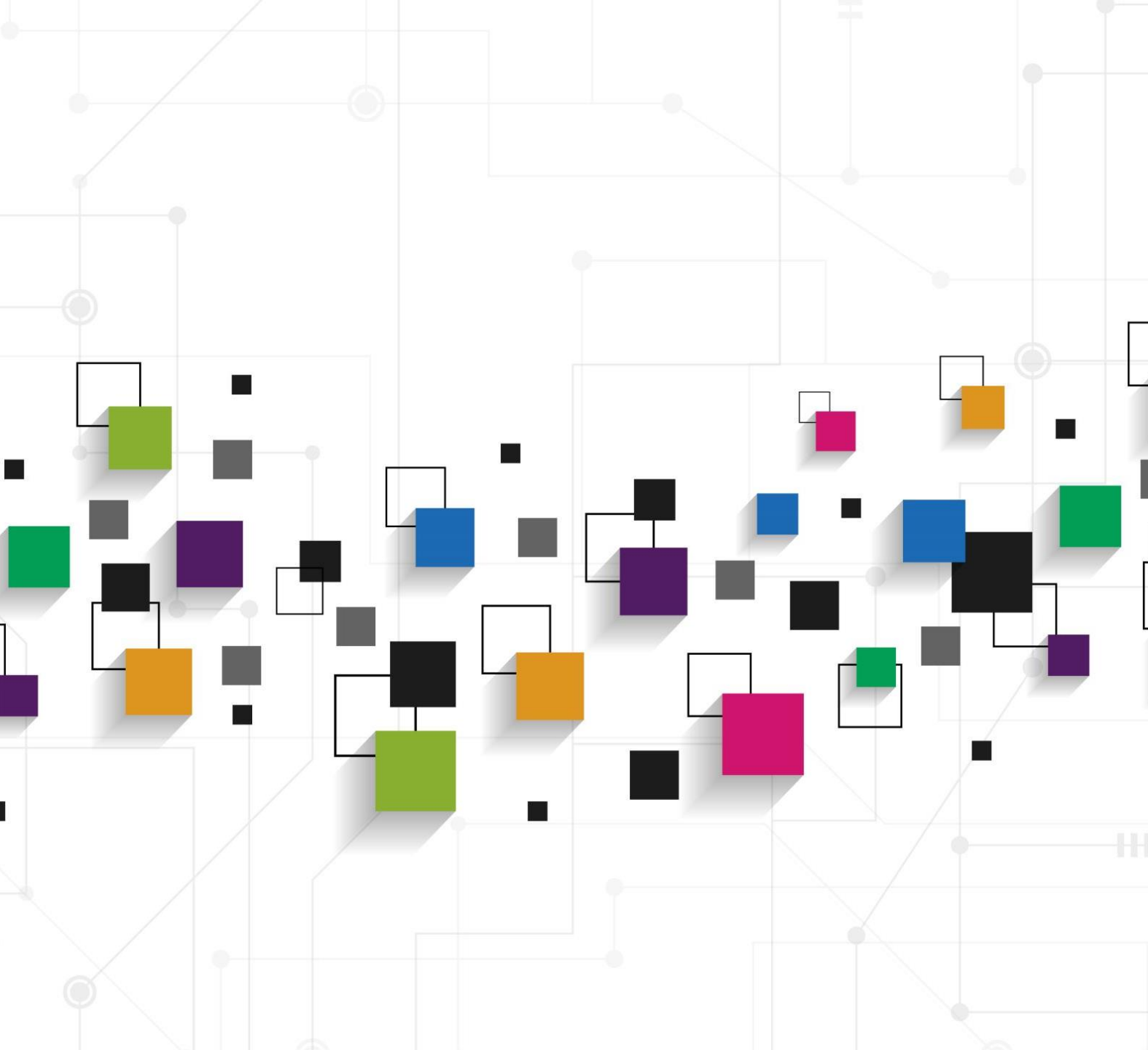


**Web and Mobile Apps**

---

WHAT ARE  
COMPONENTS  
REQUIRED TO  
RUN WEB APPS ?





---

## FRONT-END

- **HTML (Hypertext Markup Language):** It defines the structure and content of web pages.
- **CSS (Cascading Style Sheets):** Used for styling and layout, it controls the presentation and visual design of the web application.
- **JavaScript:** Provides interactivity and dynamic behavior to web pages. Common JavaScript libraries and frameworks include React, Angular, and Vue.js.
- **Responsive Design:** Ensures that the web application is accessible and looks good on various devices and screen sizes.





## BACK-END

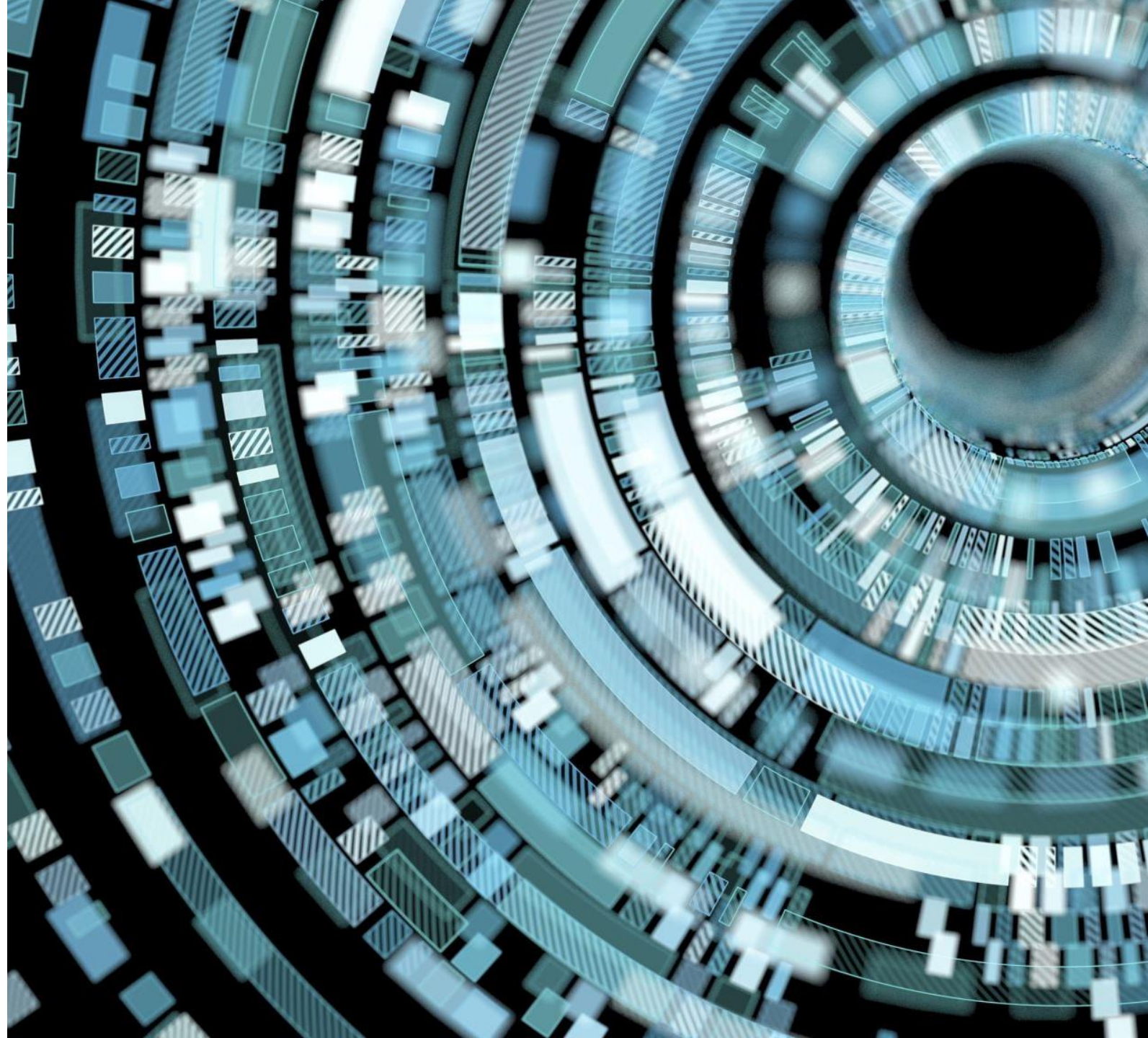
- **Server:** The back-end server handles requests from clients, processes data, and returns responses. It often runs on a web server framework or application server.
- **Application Logic:** This component contains the core functionality of the web app. It includes business logic, data processing, and communication with databases and external services.
- **Database:** Stores and manages data used by the application. Common databases include MySQL, PostgreSQL, MongoDB, and SQL Server.
- **APIs (Application Programming Interfaces):** Provide a way for the front-end and back-end to communicate. APIs define the rules and formats for requests and responses.



---

# MIDDLEWARE

- **Authentication and Authorization:** Middleware components handle user authentication and authorization, ensuring that only authorized users can access certain features or data.
- **Caching:** Caches data or responses to improve application performance.
- **Logging and Error Handling:** Logs events and errors for debugging and monitoring purposes.
- **Communication and Transaction Handling:** Handling communication in the case of complex systems and making communications reliable across components.





# DEMONSTRATIONS

