



# JAVA FULL STACK

• with AI Specialization •

FOUNDATION PROGRAM



contact@regexsoftware.com



www.regexsoftware.com

# ABOUT THE PROGRAM



At Regex Software, we offer a comprehensive 18–24 Month Java Full Stack & AI Specialization Program designed especially for B.Tech, BCA, B.Sc, BBA, B.Com and other college students who want to build a strong career in the tech industry.

This program is ideal for 1st and 2nd year students who want to start learning future technologies early and gain a competitive edge in the industry. The course is designed to build strong fundamentals first and then gradually move toward advanced concepts in Java Full Stack Development with AI, Machine Learning, and related technologies.

At Regex Software, we believe in practical learning rather than only theoretical knowledge. Our training approach combines conceptual understanding with hands-on implementation through real-world projects, live datasets, industry case studies, and portfolio-building assignments. Throughout the 18–24 month journey, students continuously work on practical tasks that help them understand how technologies are used in real companies and business environments.

The curriculum covers Core Java, Advanced Java, Spring Boot, Microservices, Hibernate, REST APIs, HTML, CSS, JavaScript, React, Database Management, and DevOps basics along with AI & ML fundamentals. Students also learn Data Structures, OOPs, System Design basics, Git & GitHub, Testing, and Deployment concepts. The entire learning path is designed according to current industry requirements so students become job-ready with practical industry skills.



## MODE

Physical  
(Jaipur & Ahmedabad)  
or Online  
(Google Meet)



## DURATION

18–24 months  
+ 6 months support



## PARTICIPANTS

18 - 20 Per  
Batch

## WHAT YOU WILL LEARN ?



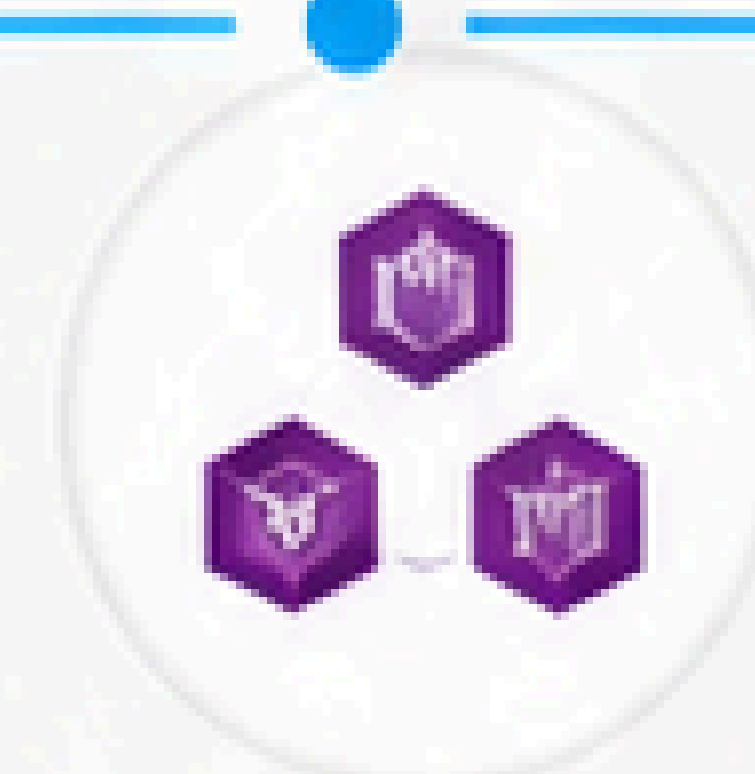
Java



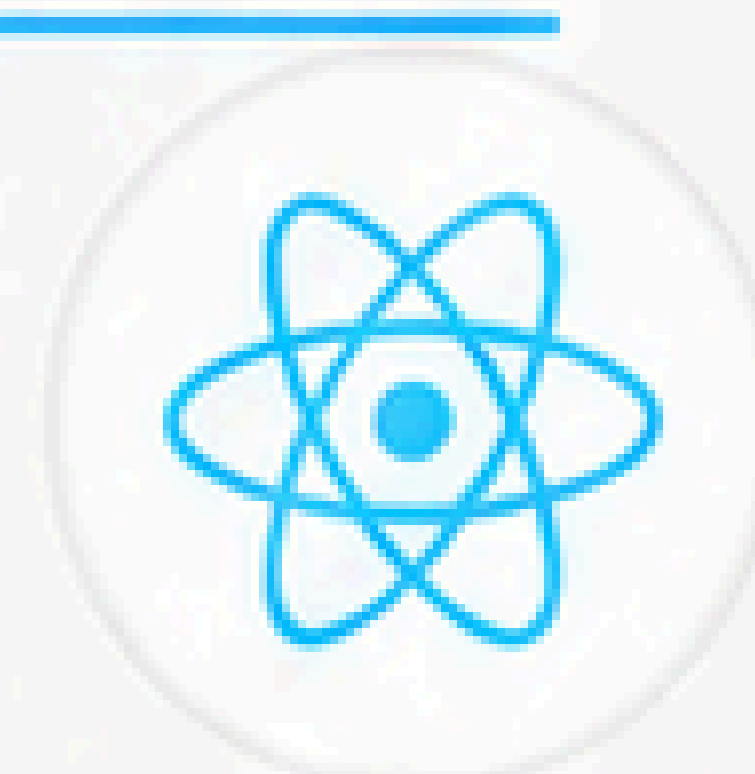
DSA



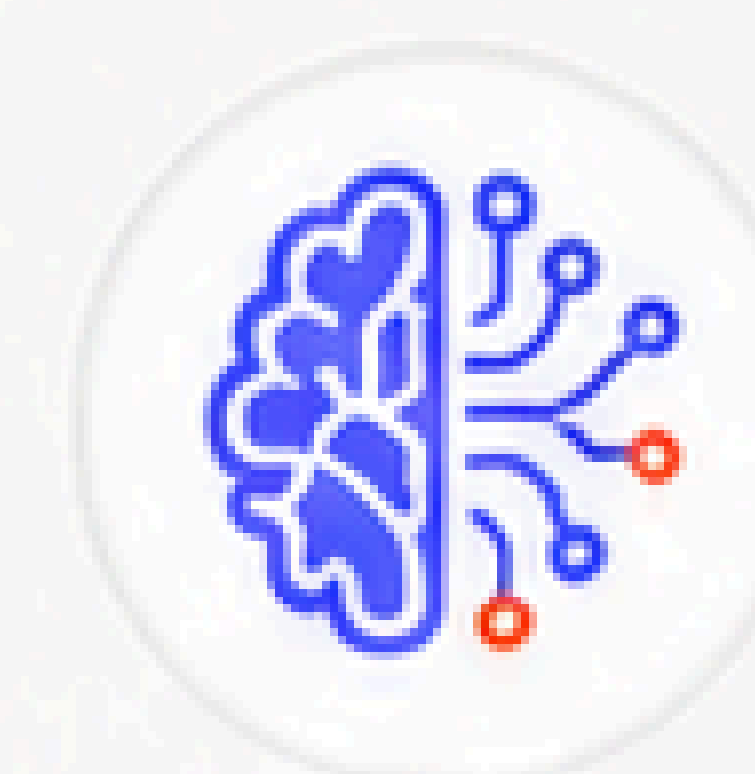
Spring Boot



Microservices



React



AI / ML



SQL



Git & GitHub



Docker



Kubernetes



Jenkins



Postman



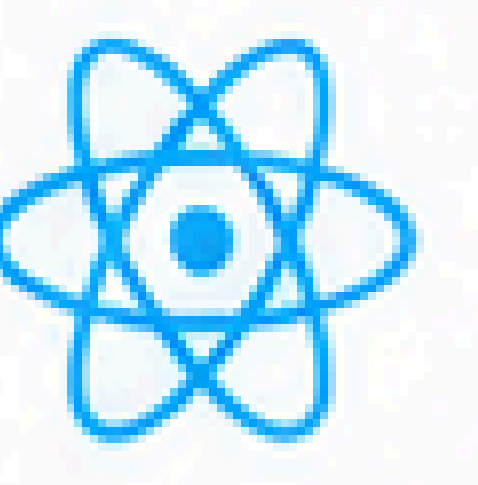
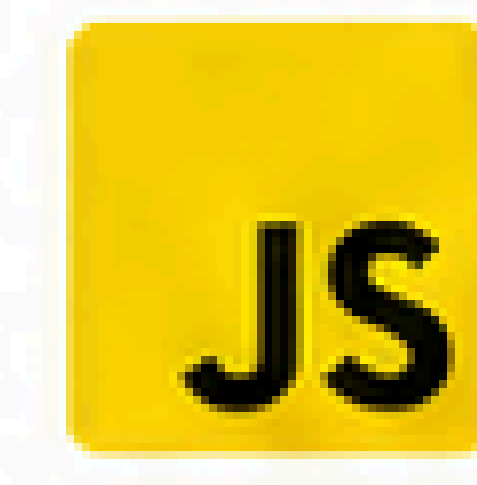
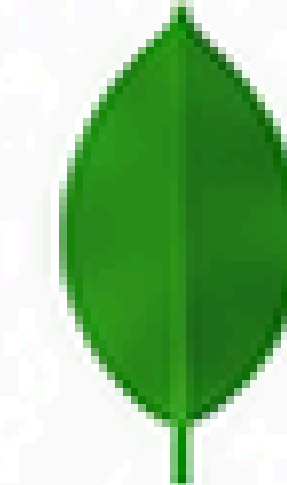
Swagger



# JAVA FULL STACK

with  
**AI Specialization**

## FOUNDATION PROGRAM



Spring Boot

MongoDB

HTML

CSS

JavaScript

React



### STUDY MATERIAL

- E-Notes
- Assignments & PDF test
- Live Video Lectures
- Access of Recordings & Study Material
- Mentorship Support
- Work on multiple Minor Projects & Use Cases
- Work on Live Projects



### OUTPUT

- Build modern, responsive and scalable web applications
- Become job-ready Full Stack Developer with real-world projects
- Understand working of Java Full Stack in depth
- Build projects on multiple domains
- Work on more than 25 Use CASES & Projects
- Learn to deploy your applications on various cloud platforms
- Learn Git, GitHub, REST APIs, Authentication, and DevOps basics

## PACKAGE OFFERED SO FAR

### IT CANDIDATES



Minimum Package  
4 LPA



Average Package  
4 – 6 LPA



Overall Highest Package  
39 LPA

### NON-IT CANDIDATES



Minimum Package  
3 LPA



Average Package  
3.5 – 5 LPA



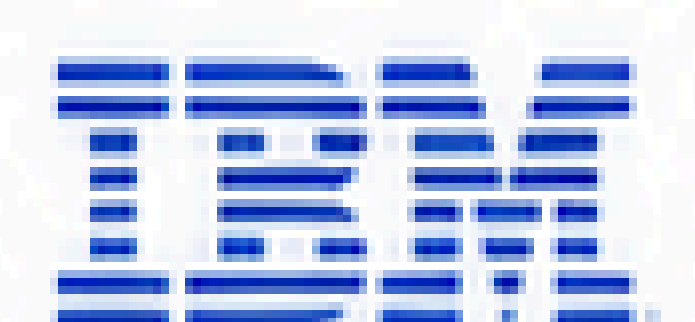
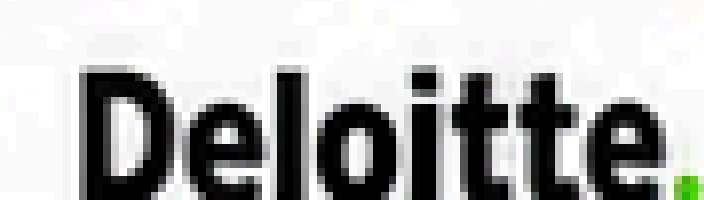
Overall Highest Package  
14.5 LPA



### EXTRA SESSIONS:

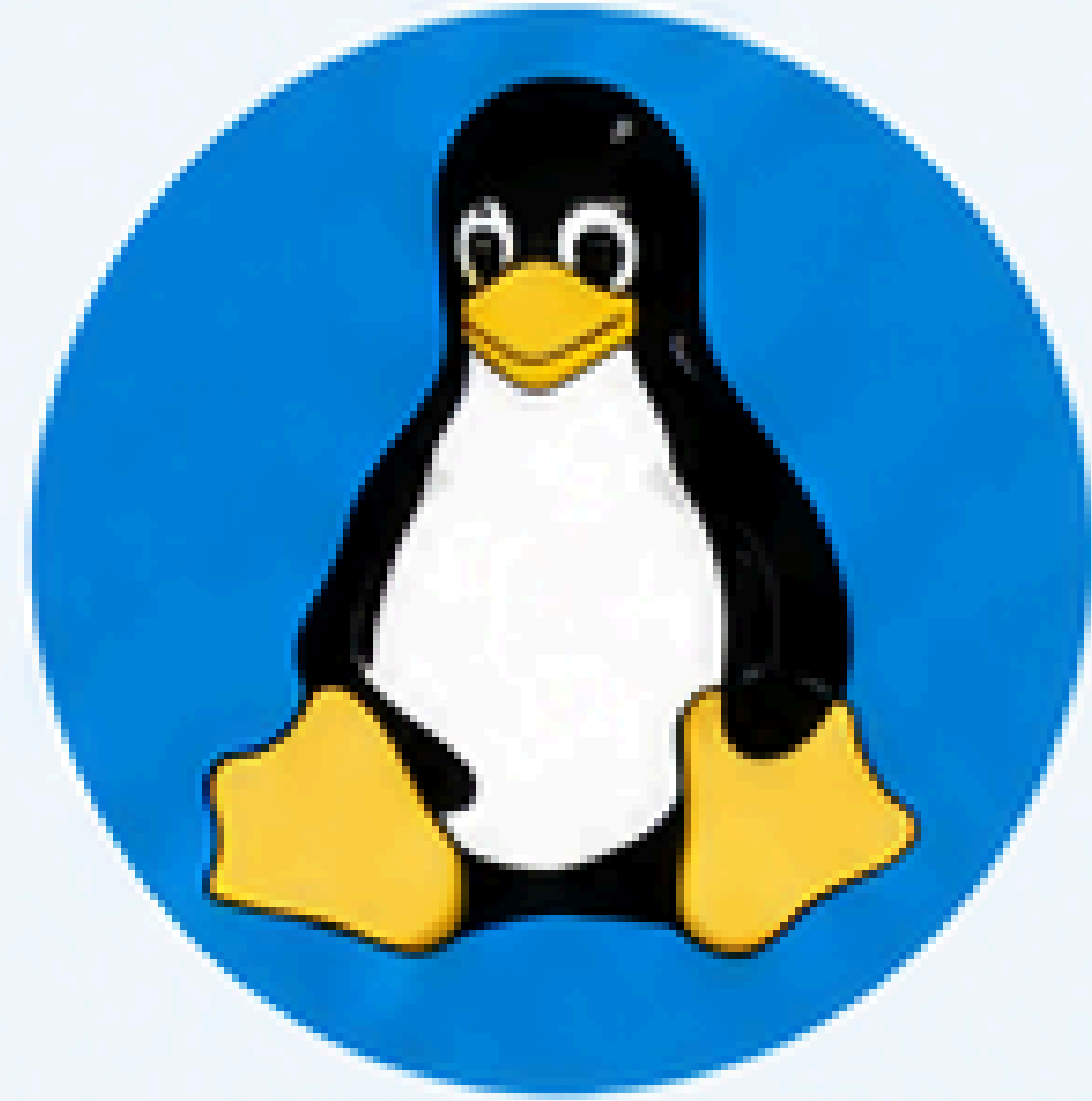
Additional Session on Git, Docker, AWS Basics, Jenkins and many more for all students.

### OUR STUDENTS PLACED // PARTNERSHIP



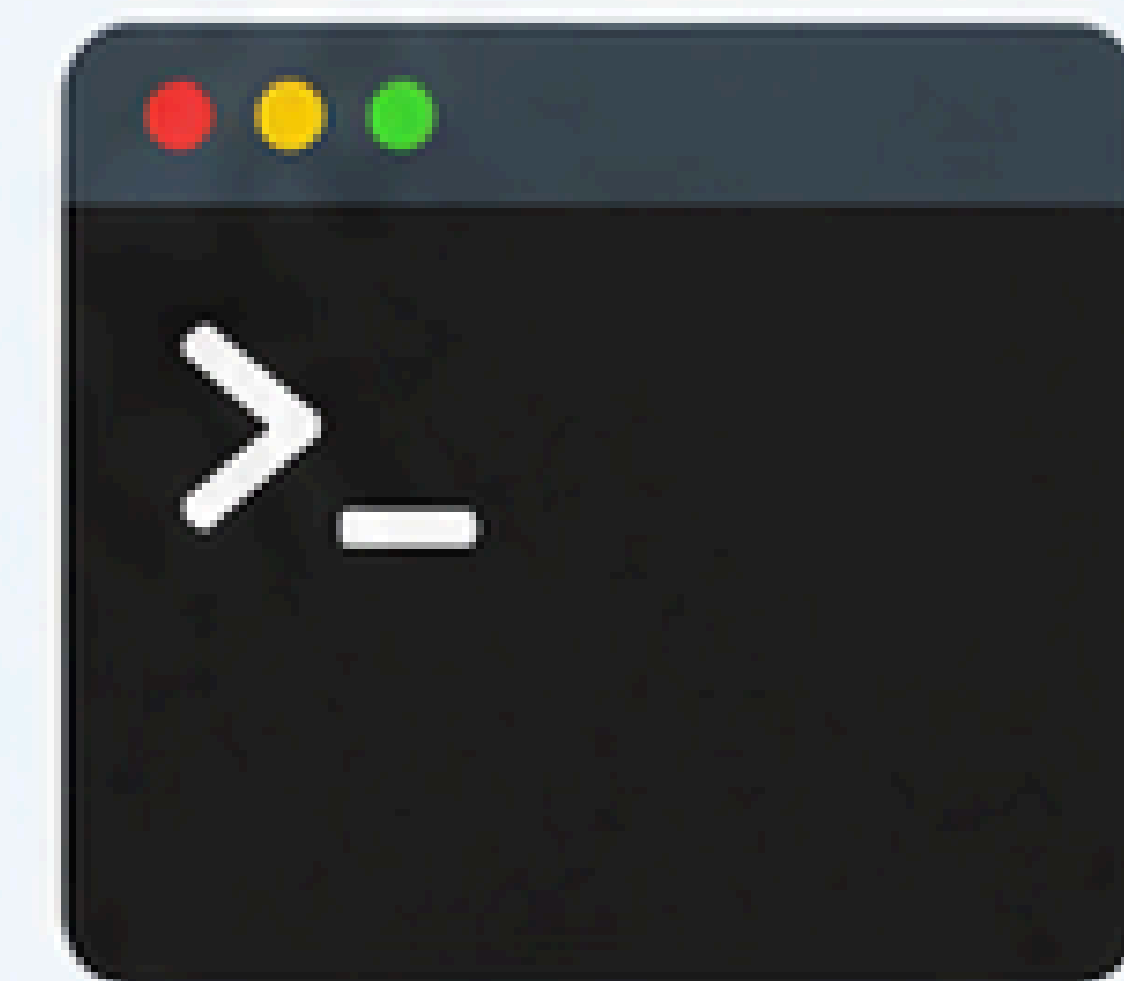
# COURSE CONTENT:

## PHASE 1



### 1. LINUX BASICS

- Introduction to Linux
- Basic Commands of Linux OS
- Vi Editor
- Tar Archive
- User Management and Permission



### 2. GITHUB

- Git vs GitHub
- Git Commands
- Git Branches
- Branching and Merging
- Git Push vs Pull Commands



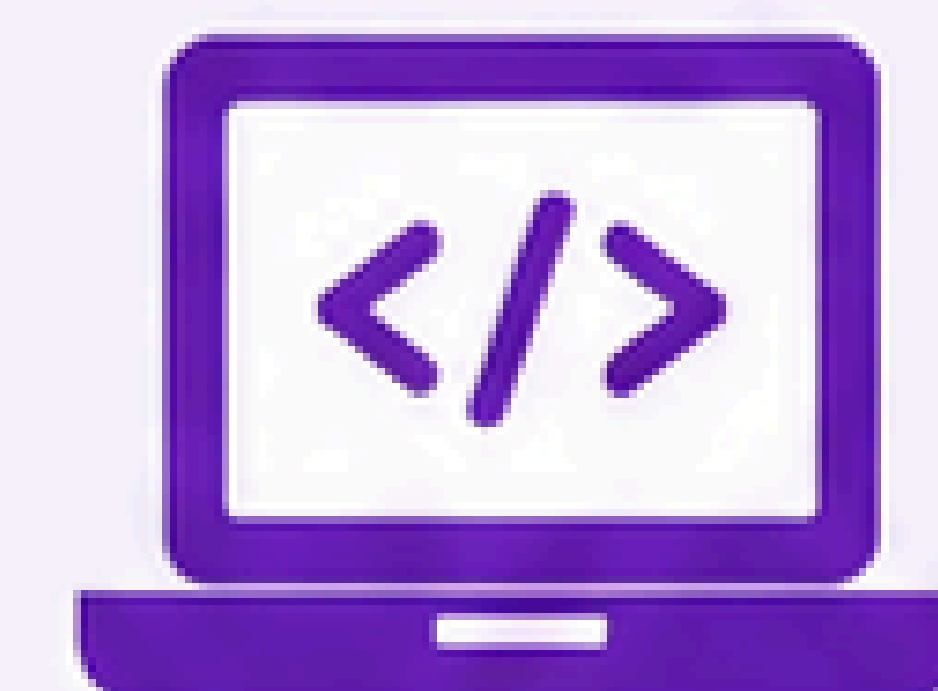
### 3. C PROGRAMMING – CORE CONCEPTS

- Data Types
- Variables
- Variable Scope (Local, Global)
- Constants
- Operators
- Decision Making Statements (if Statement, if..else, Switch)
- Loops (While Loop, Do-While Loop, for-loop)



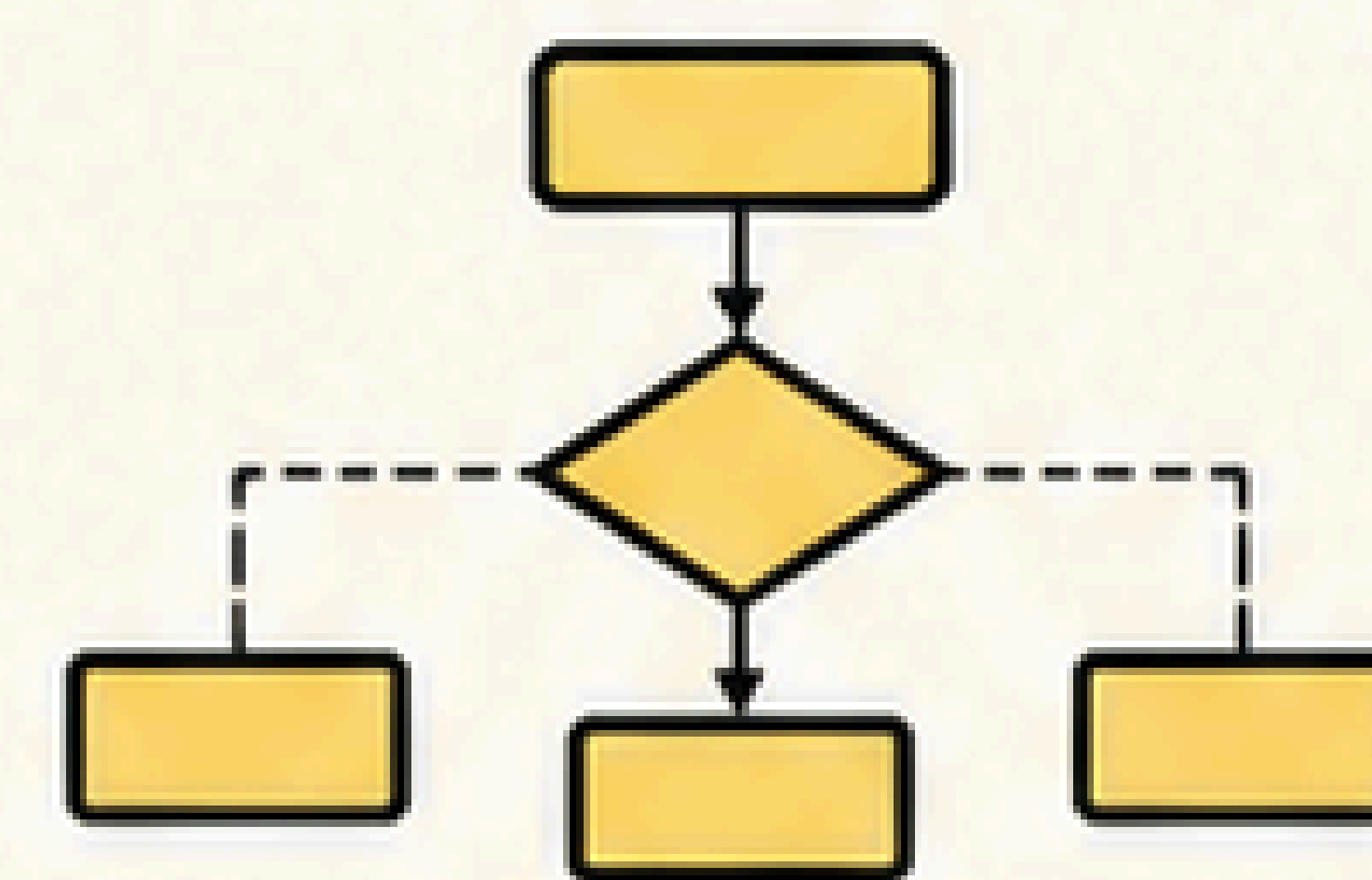
### 4. C PROGRAMMING – ADVANCED BASICS

- While Loop
- Do-While Loop
- for-loop
- Basic I/O Functions



### 5. LOGIC BUILDING

- Logical Thinking and Approach
- Flowcharts and Pseudocode
- Pattern Recognition
- Basic Problem-Solving



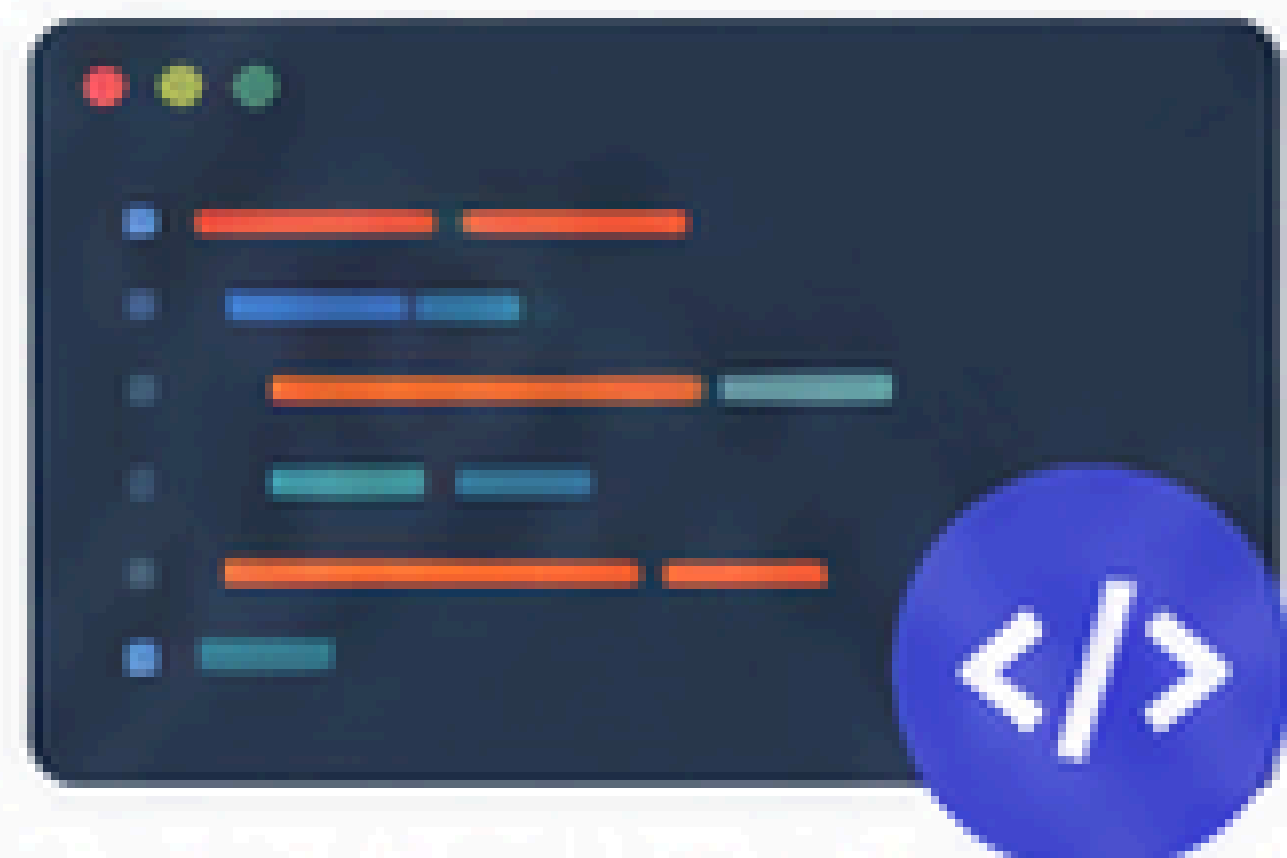
### 6. BASICS PROBLEM-SOLVING

- Understanding Problem Statements
- Breaking Problems into Steps
- Test Cases and Debugging
- Optimizing Solutions (Basics)



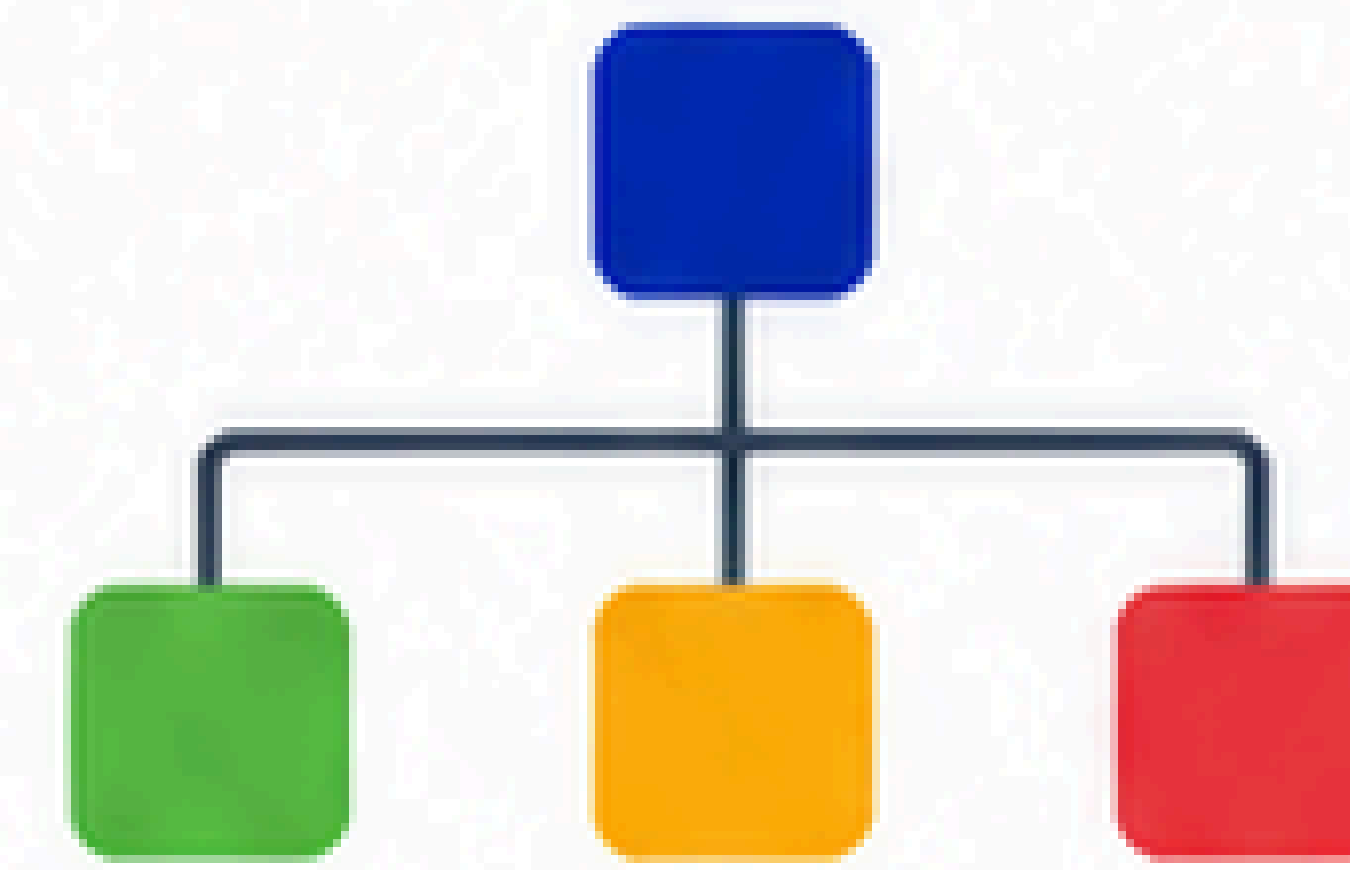
# PHASE 2

## PART 1 – JAVA BASICS



- Java
- Java Object Class
- Java OOPs Concepts
- AOP
- Naming Convention
- Object and Class
- Method
- Constructors
- static keyword
- this keyword
- Java Inheritance
- Inheritance (IS-A)
- Aggregation (HAS-A)

## PART 2 – JAVA OOPS ADVANCED



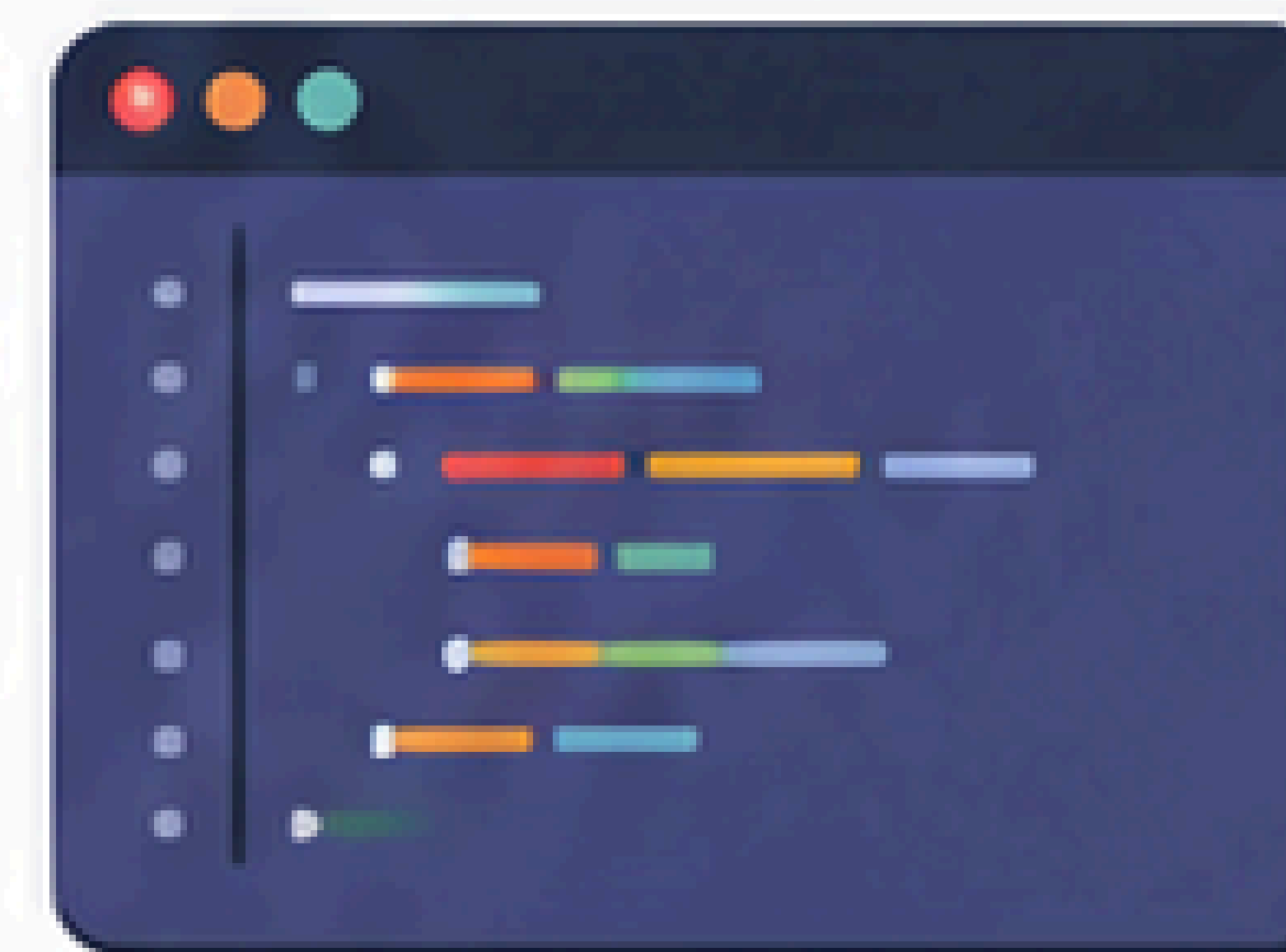
- Java
- Java Polymorphism
- Method Overloading
- Method Overriding
- Covariant Return Types
- super keyword
- Instance Initializer block
- final keyword
- Runtime Polymorphism
- Dynamic Binding
- instanceof operator

## PART 3 – JAVA ABSTRACTION & ENCAPSULATION



- Java
- Java Abstraction
- Abstract class
- Interface
- Abstract vs Interface
- Java Encapsulation
- Package
- Access Modifiers
- Encapsulation

## PART 4 – ARRAYS & PROBLEM SOLVING



- Java
- Java Arrays
- 1 D and 2 D array
- Hacker rank
- Leetcode
- codechef

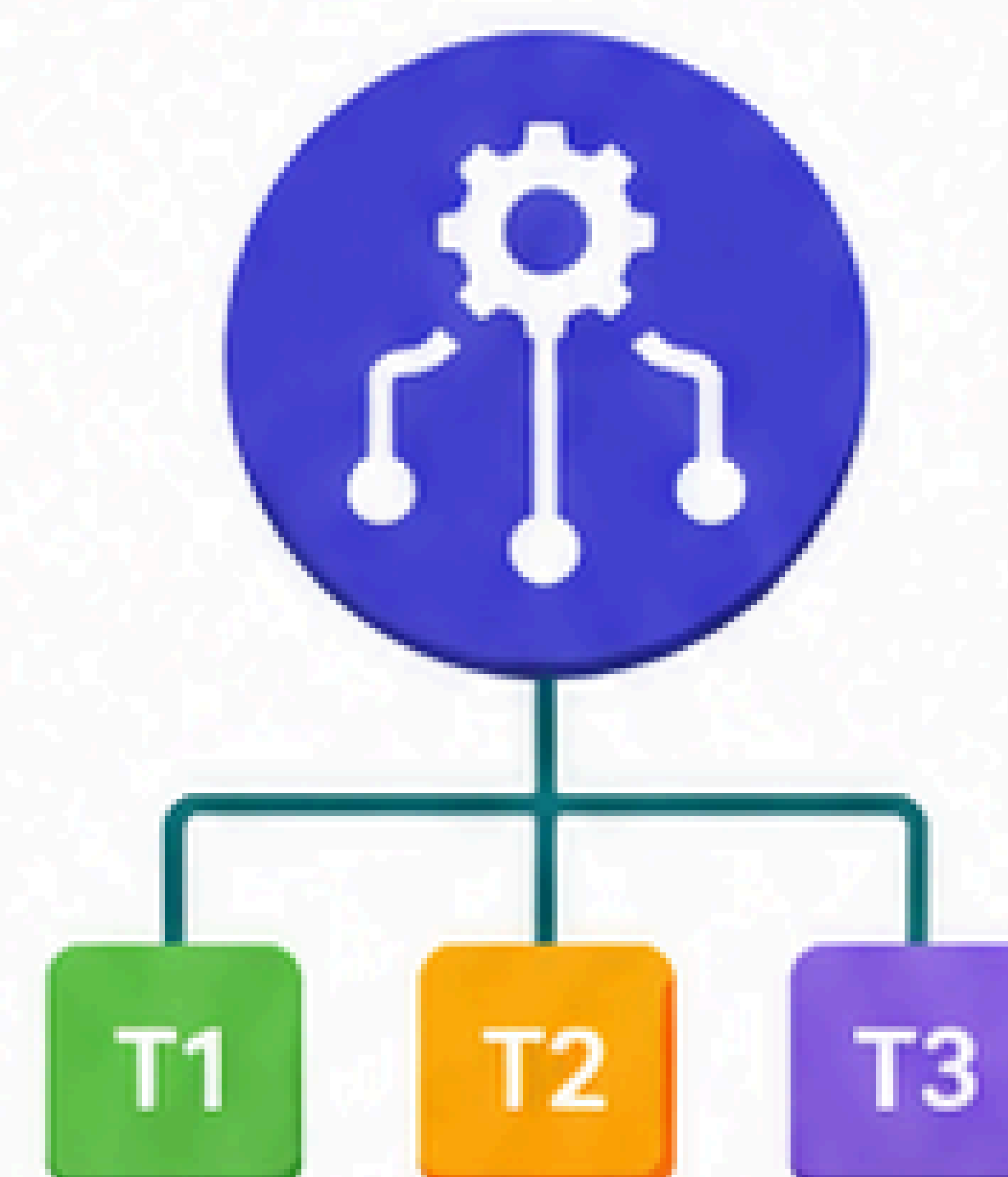
## PART 5 – EXCEPTION HANDLING



```
try {  
    // code  
} catch (Exception e) {  
    // handle  
} finally {  
    // cleanup  
}
```

- Java
- Exception Handling
- Java Exceptions
- Java Try-catch block
- Multiple Catch Block
- Java Nested try
- Java Finally Block
- Java Throw Keyword
- Java Exception Propagation
- Java Throws Keyword
- Throw vs Throws
- Final vs Finally vs Finalize
- Exception Handling with Method Overriding
- Custom Exceptions

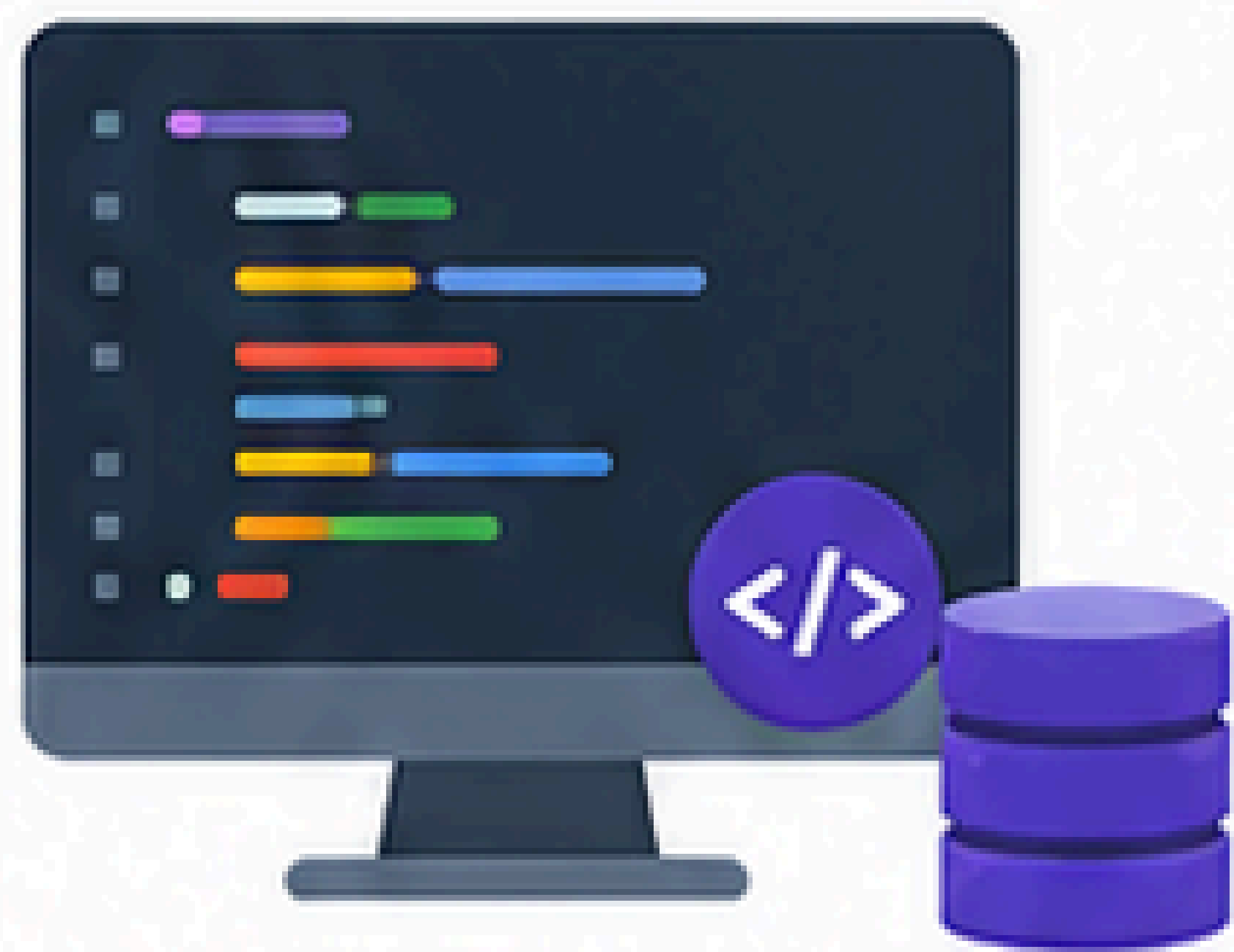
## PART 6 – MULTITHREADING



- Java
- Java Multithreading
- What is Multithreading
- Life Cycle of a Thread
- How to Create Thread
- Thread Scheduler
- Sleeping a thread
- Start a thread
- twice Calling run() method
- Joining a thread
- Naming a thread
- Thread Priority
- Daemon Thread
- Thread Pool
- Thread Group
- Shutdown Hook
- Garbage Collection & Runtime class
- Java Synchronization
- Synchronization in java
- synchronized block
- static synchronization
- Deadlock in Java

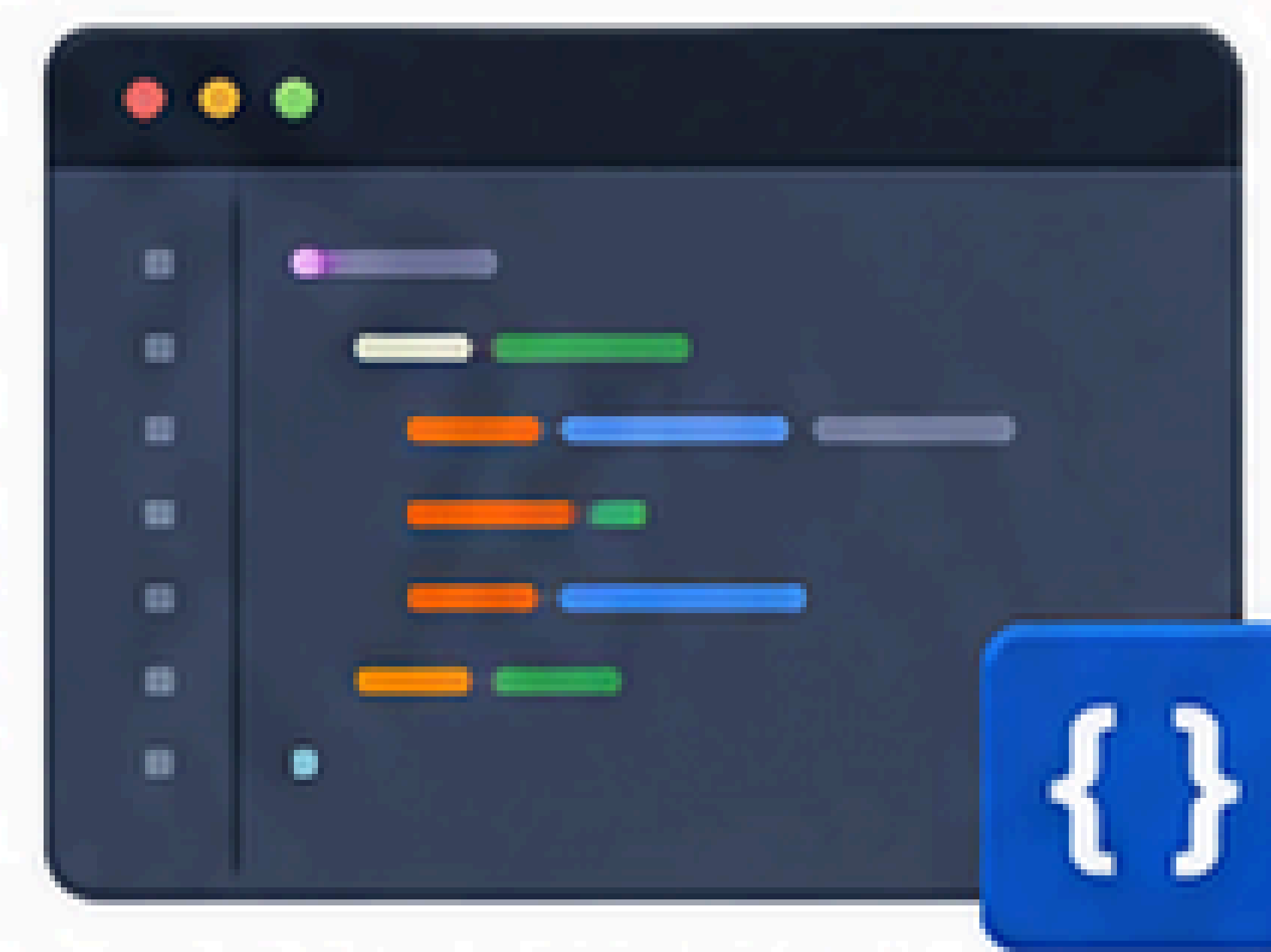
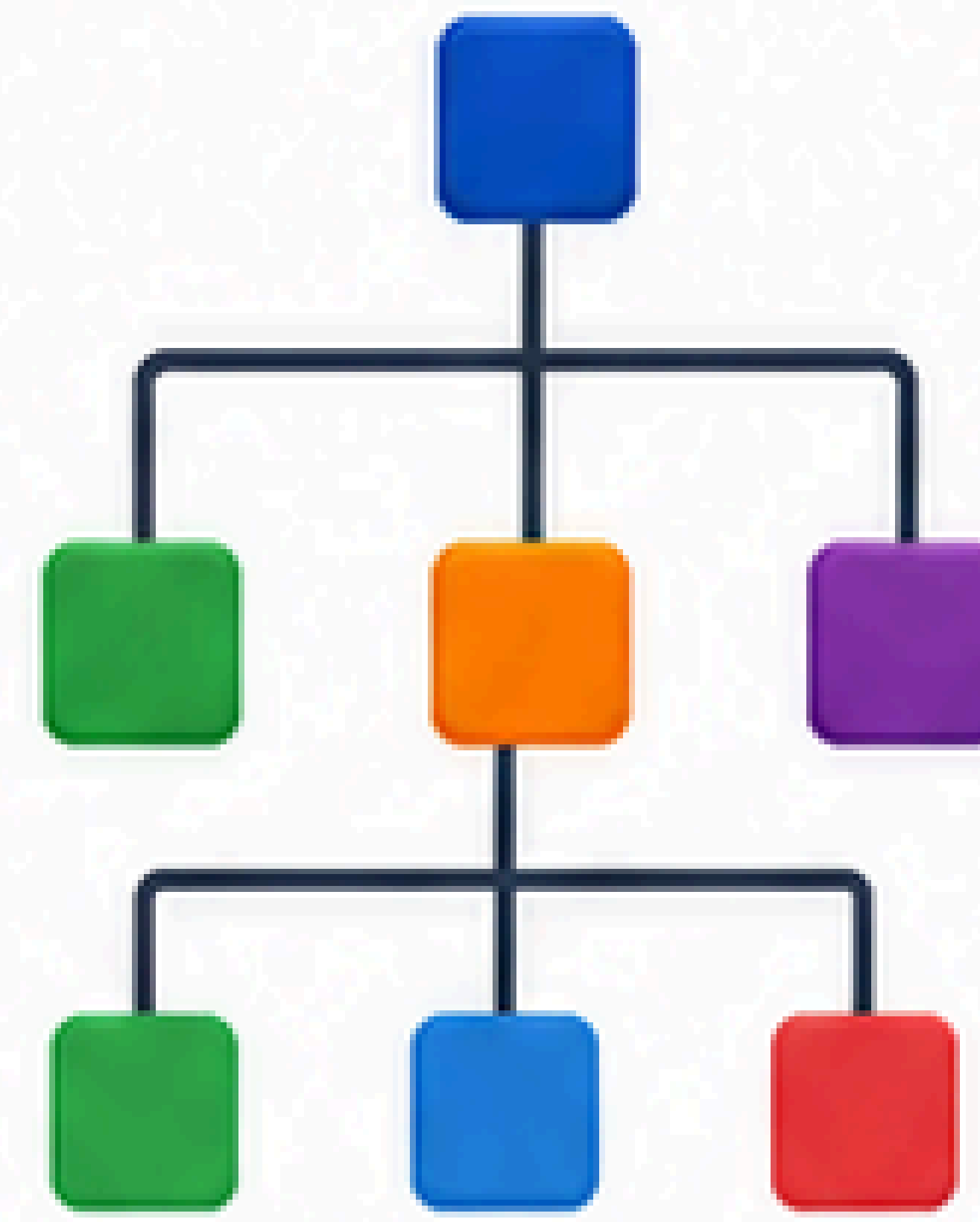
# PHASE 3

## PART 1 – JAVA COLLECTIONS BASICS



- Java Collections
- Collection Framework
- Java ArrayList
- LinkedList
- ArrayList vs LinkedList
- List Interface
- HashSet
- LinkedHashSet
- TreeSet
- Queue & PriorityQueue
- Deque & ArrayDeque
- Map Interface
- HashMap

## PART 2 – ADVANCED COLLECTIONS



- HashMap
- Working of HashMap
- Java LinkedHashMap
- Java TreeMap
- Java Hashtable
- HashMap vs Hashtable
- Java EnumSet
- Java EnumMap
- Collections class
- Sorting Collections
- Comparable interface
- Comparator interface
- Comparable vs Comparator
- Properties class
- ArrayList vs Vector
- Vector
- Stack
- Collection Interface
- Iterator Interface
- Deque Interface
- ConcurrentHashMap

## PART 3 – DATE, TIME & UTILITIES



- Java 8 Date Time
- Java Date Time API
- Java LocalDate
- Java LocalTime
- Java LocalDateTime
- Java MonthDay
- Java OffsetTime
- Java OffsetDateTime
- Java Clock
- method , recursion
- file handling
- regex

## PART 4 – SORTING & ALGORITHM PROBLEMS



- sorting algorithms
- bubble sort
- merge sort
- quick sort
- heap sort
- count sort
- problem solving on algorithms

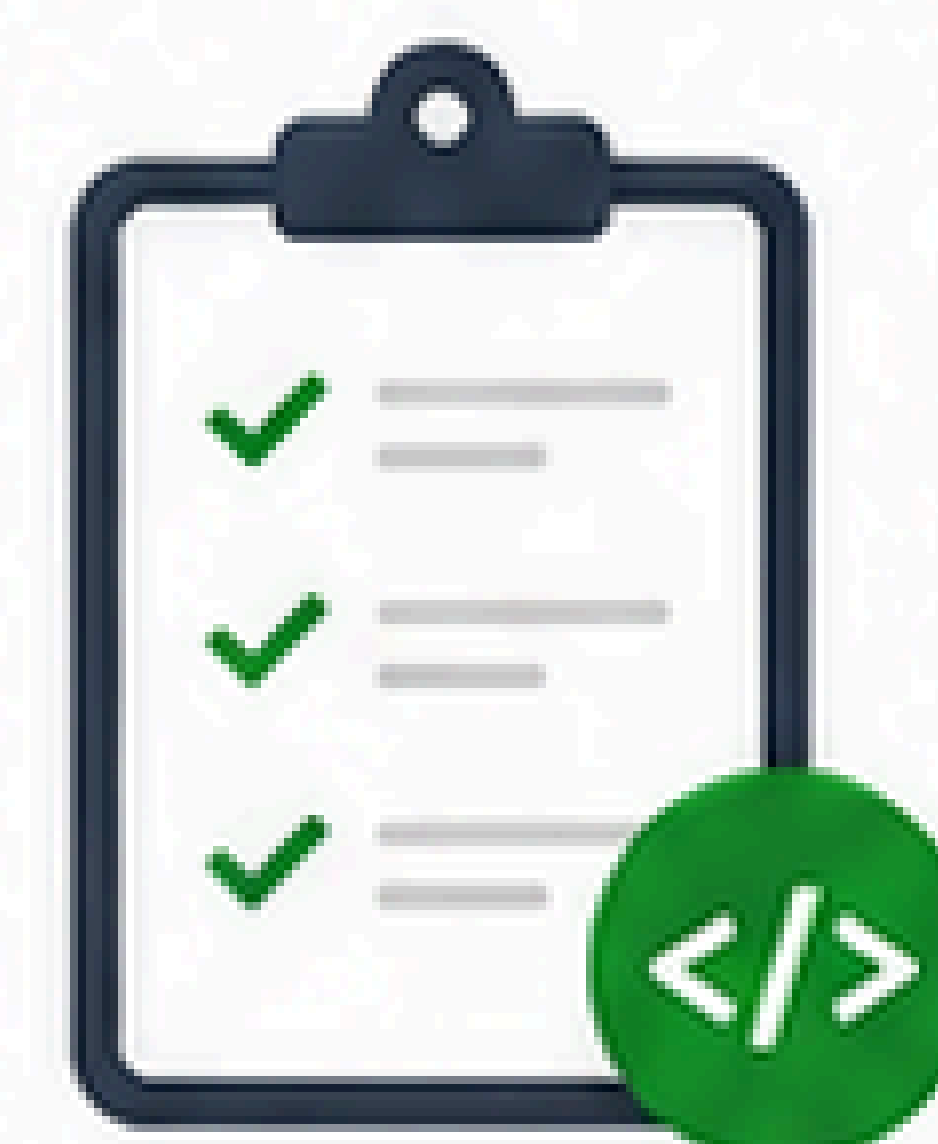
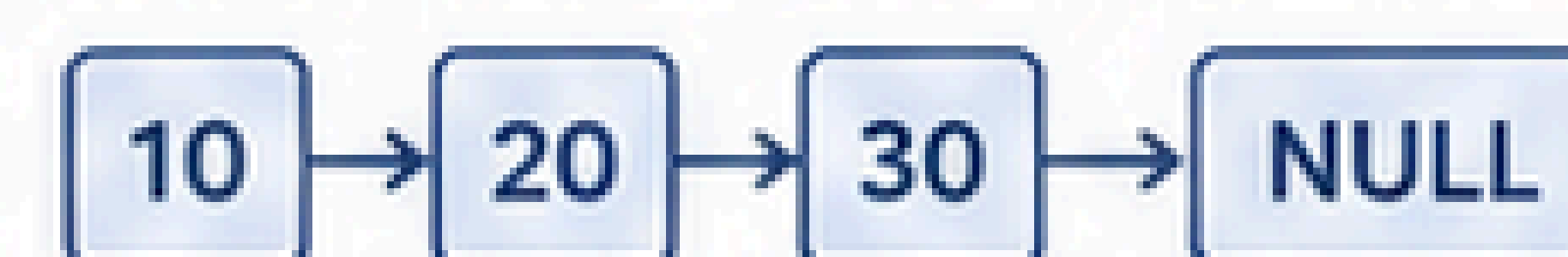
## PART 5 – STRING HANDLING

“String”

```
String str = "Hello Java";  
StringBuilder sb =  
    new StringBuilder(str);  
sb.append(" Developer");
```

- string and String Builder
- String Buffer
- method of string
- problem solving on string

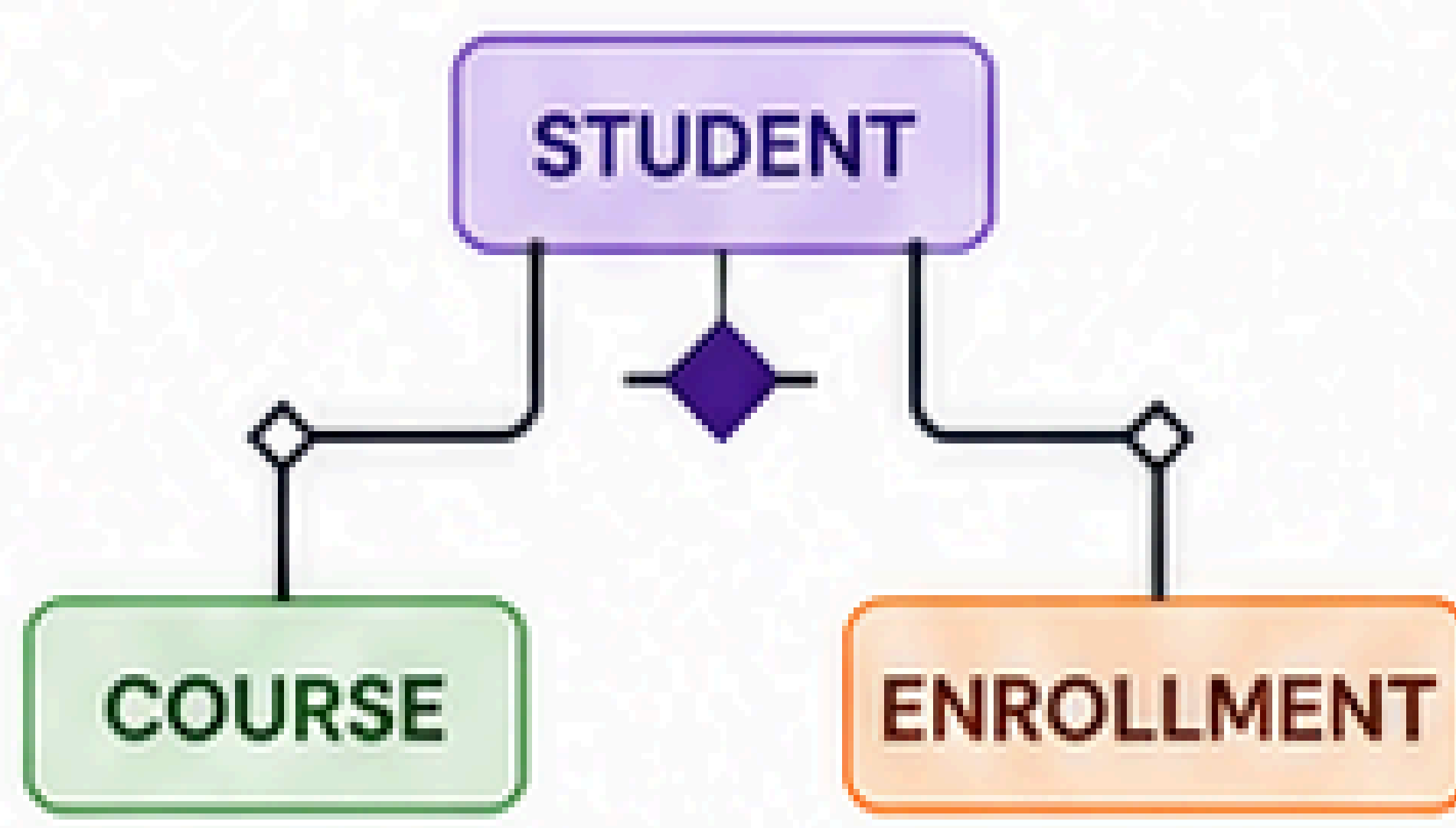
## PART 6 – DATA STRUCTURES & PRACTICE



- Java linkedlist
- insertion
- deletion
- updatation
- problem solving questions
- leetcode
- hacker rank
- codechef

# PHASE 4

## PART 1 – SQL BASICS & DATABASES



- Introduction to E-R diagram
- Understanding entity and relationship
- Working with symbols
- Type of relationship and data models
- Defining Database
- Type of databases
- DBMS vs RDBMS
- Installing MySQL database
- Type of SQL statements
- Select statement

## PART 2 – SQL QUERIES & FUNCTIONS

```
SELECT *
FROM users
WHERE age > 18
ORDER BY name;
```



- Operators in SQL
- SQL querying statement
- String, date, number functions
- Working with aggregation functions
- Group by statement
- Where vs having clause
- Sorting data in SQL
- Advanced sorting and aggregate functions

## PART 3 – SUBQUERIES & JOINS

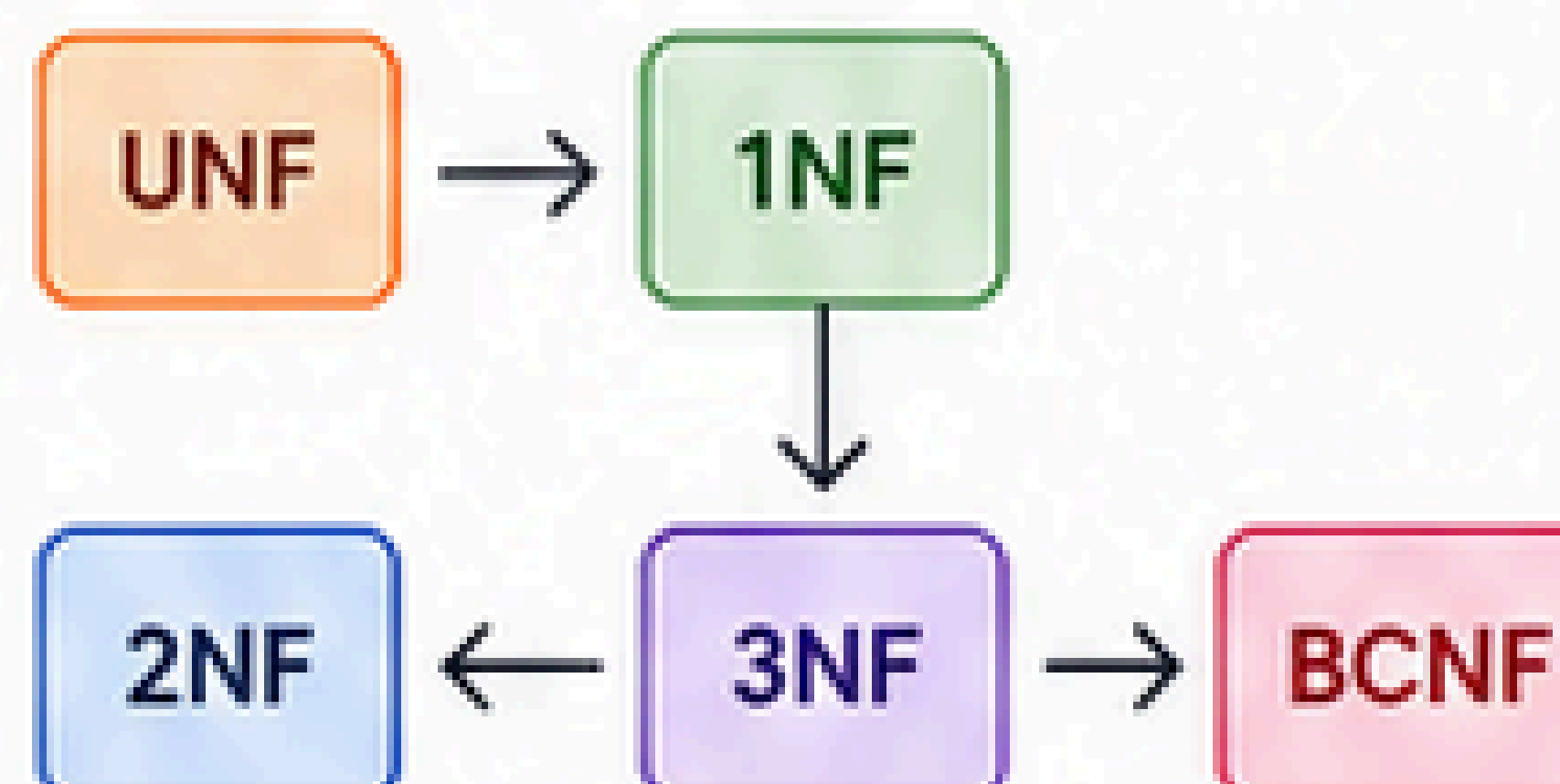
```
SELECT name
FROM students
WHERE id IN (
  SELECT student_id
  FROM marks
);
```

STUDENTS		COURSES	
id	name	id	title
1	A	1	Java
2	B	2	SQL
3	C	3	Python

- Subquery
- Type of subqueries
- Advanced subquery
- Joins
- Understanding need of joins
- Inner join
- Outer joins
- Self join, cross join, natural joins
- Set operations in SQL

## PART 4 – CONSTRAINTS, NORMALIZATION & DDL

USERS		
id	name	email
1	John	john@email.com
2	Alice	alice@email.com
3	Bob	bob@email.com



- Constraints
- Type of keys
- Setting up the first constraint to your tables
- Normalization, types of dependencies and anomalies
- 1NF, 2NF, 3NF and BCNF
- Conversion from 1 NF to 2NF, 3NF and BCNF
- DDL statement - Create, Drop Alter Queries

## PART 5 – DML, TCL & DCL



```
INSERT INTO users (name, age)
VALUES ('John', 25);

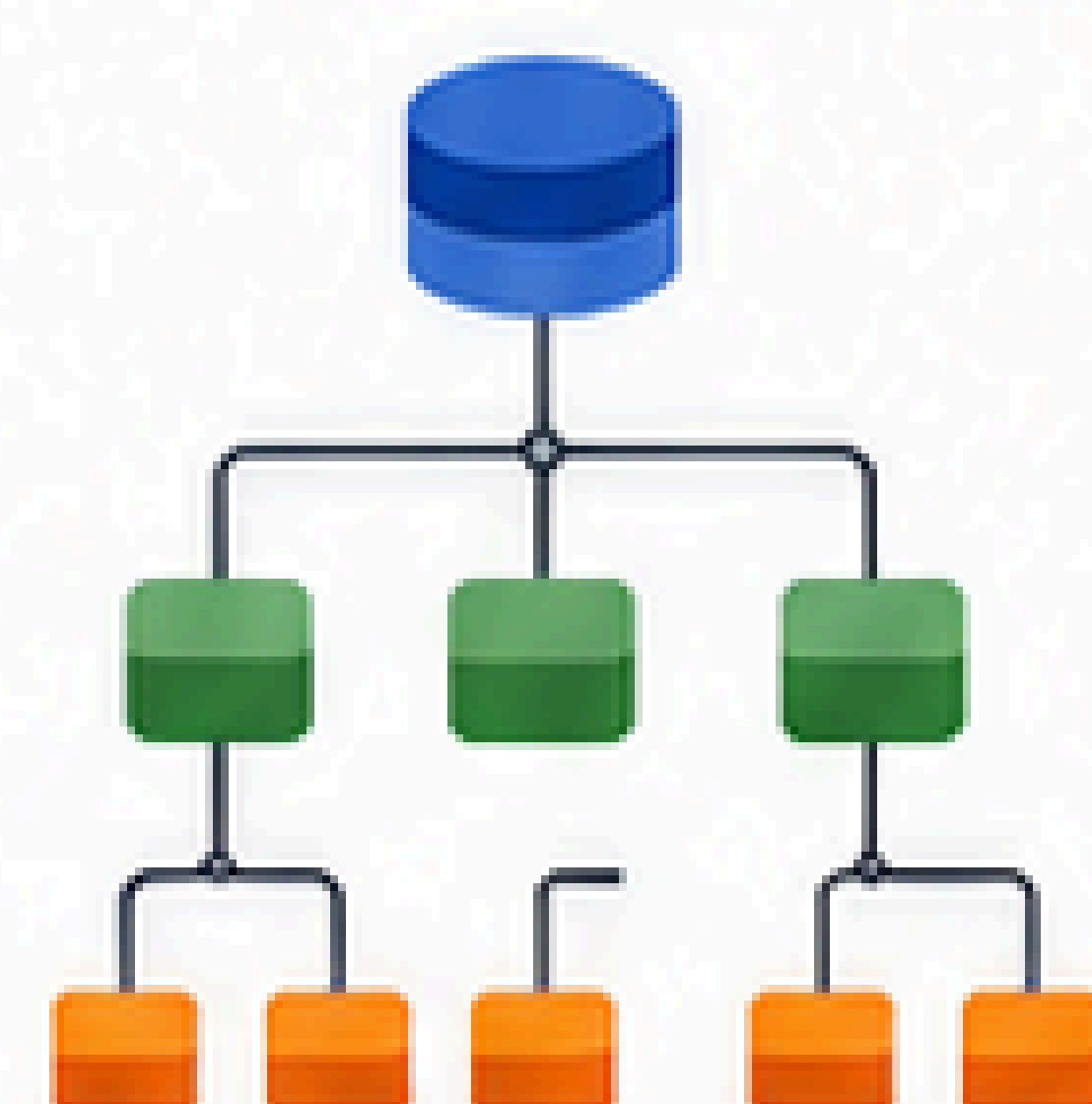
UPDATE users SET age = 26
WHERE id = 1;

DELETE FROM users WHERE id = 1;
```

- DML operations - insert, update and delete
- TCL commands - commit, rollback & savepoint
- DCL - Grant & revoke

Milestone - HackerRank SQL certificate

## PART 6 – INDEXES & OPTIMIZATION



```
CREATE INDEX idx_name
ON users(name);
```

- Indexes
- Clustered vs Non-clustered index
- Composite vs Column index
- Revision of SQL

Milestone 2 - HackerRank SQL certificate (Intermediate)

## PART 7 – ADVANCED FUNCTIONS & BADGE

```
SELECT dept,
COUNT(*) as total
FROM employees
GROUP BY dept;
```



- Analytical functions
- Windows functions
- (Partition by & Group up)

HackerRank - Gold Badge (5 Star on HackerRank)

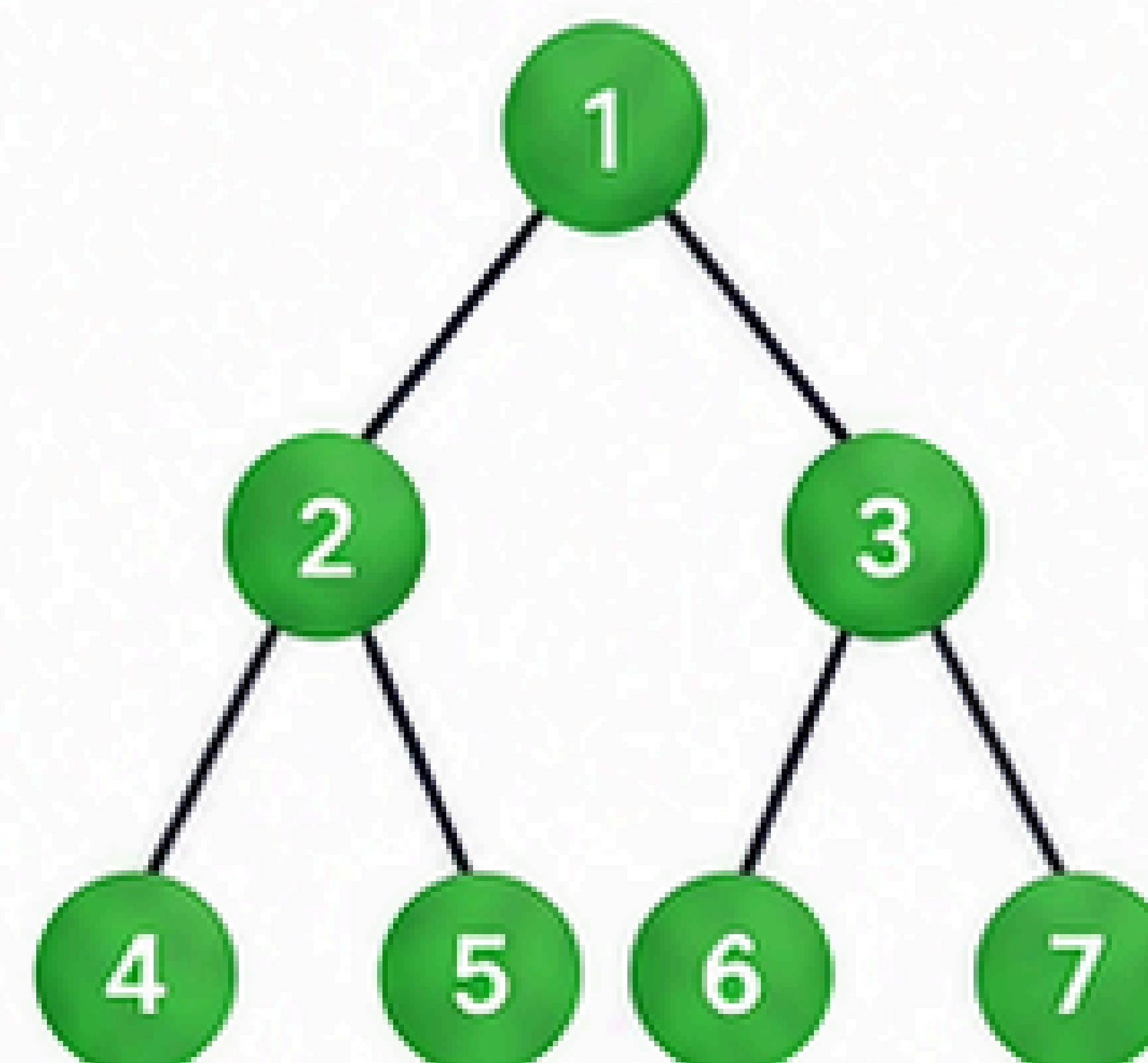
# PHASE 5

## PART 1 – QUESTIONS PLATFORM BASED



Java  
Questions Platform  
Based

## PART 2 – TREE



- tree
- create tree
- find node
- insertion
- deletion
- traversal
- pre, post and in

## PART 3 – LINKEDLIST THROUGH NODE

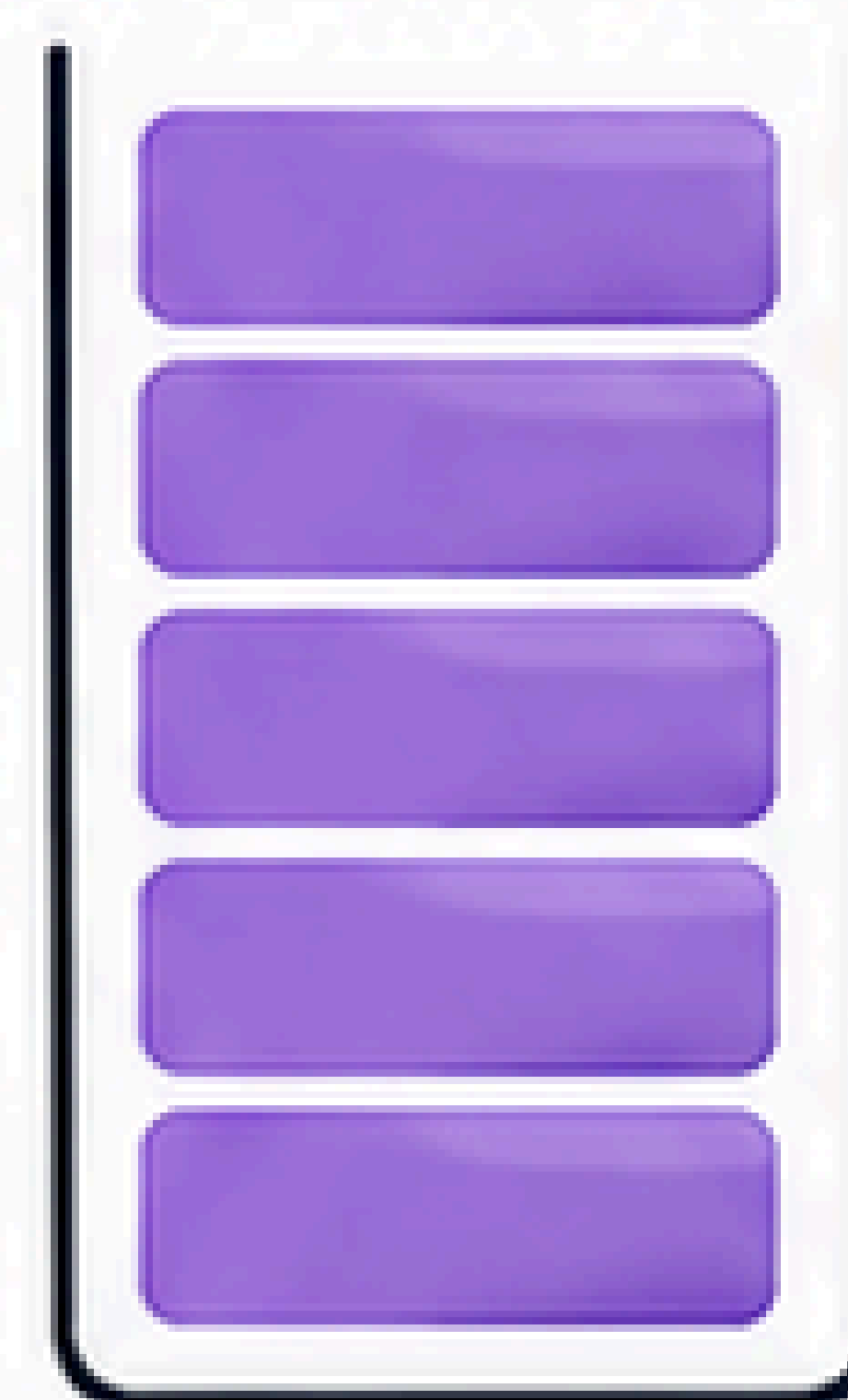


```
class Node {  
    int data;  
    Node next;  
}
```

- insertion
- deletion
- updation and mid
- element and get and set size
- problem solving questions
- leetcode
- hacker rank
- codechef

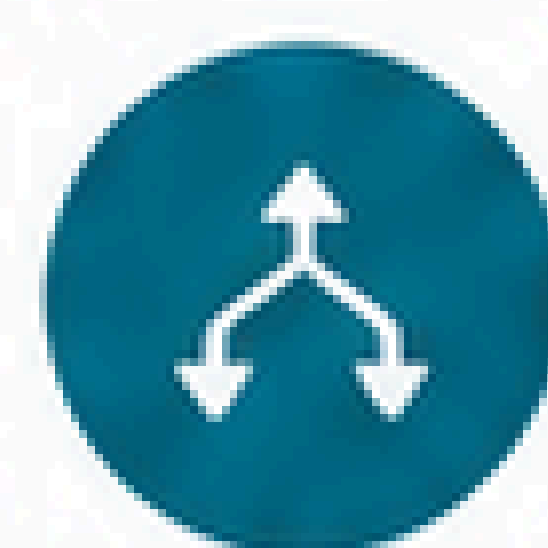
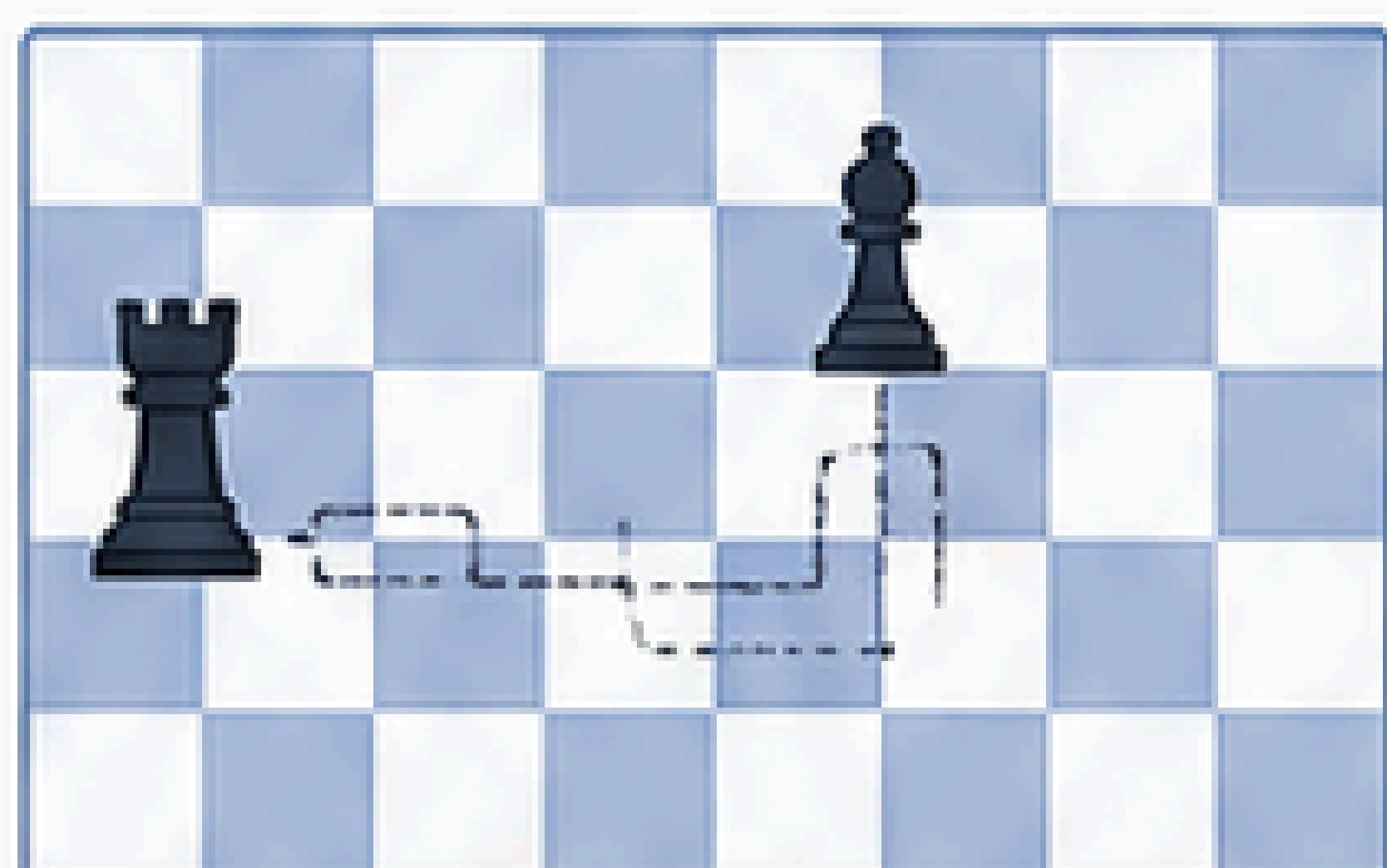
## PART 4 – STACK

TOP  
↑



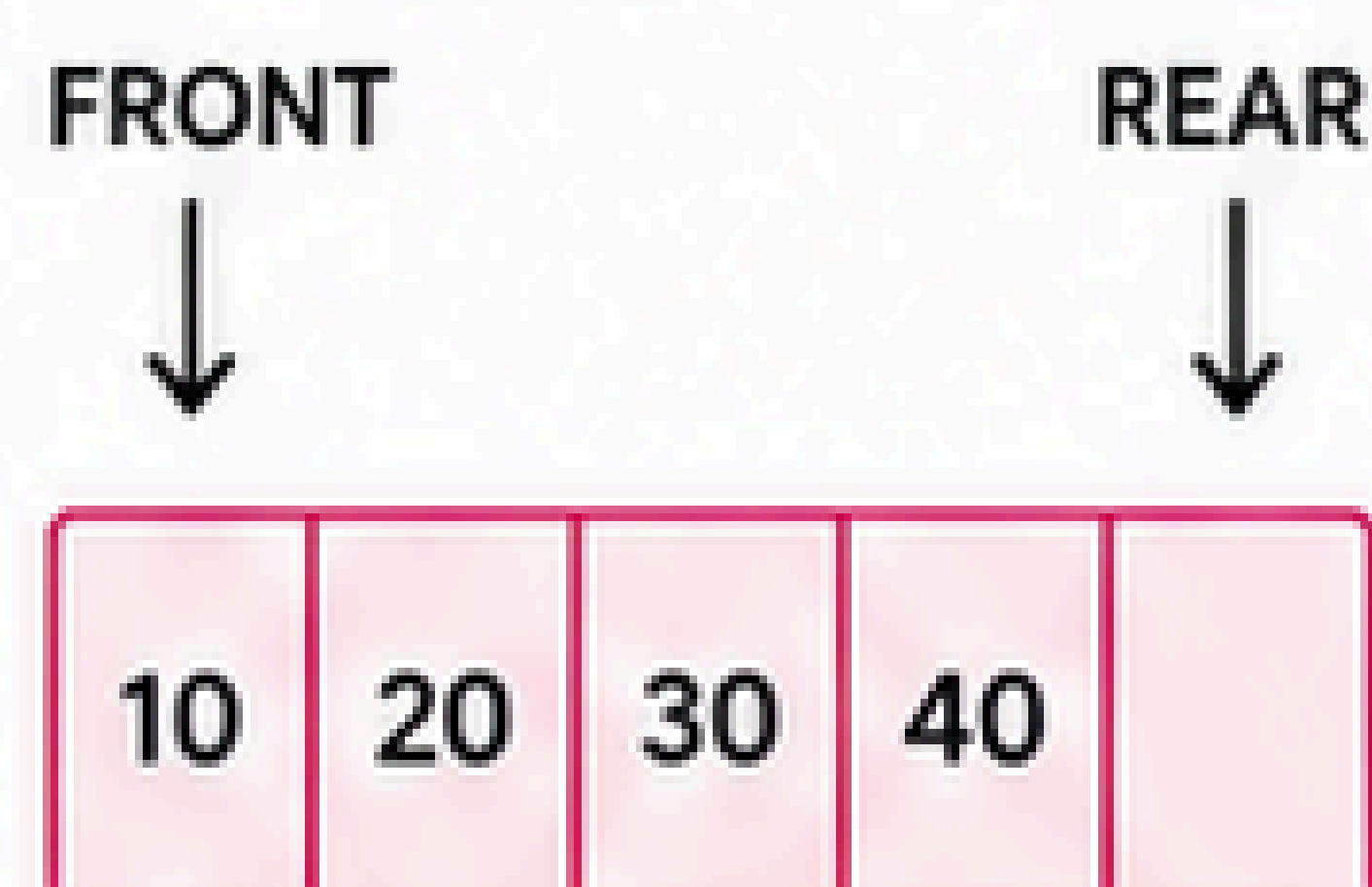
- Stack
- insertion (push)
- deletion (pop)
- updation (peek)
- problem solving questions
- leetcode
- hacker rank

## PART 5 – BACKTRACKING



Java  
backtracking

## PART 6 – QUEUE



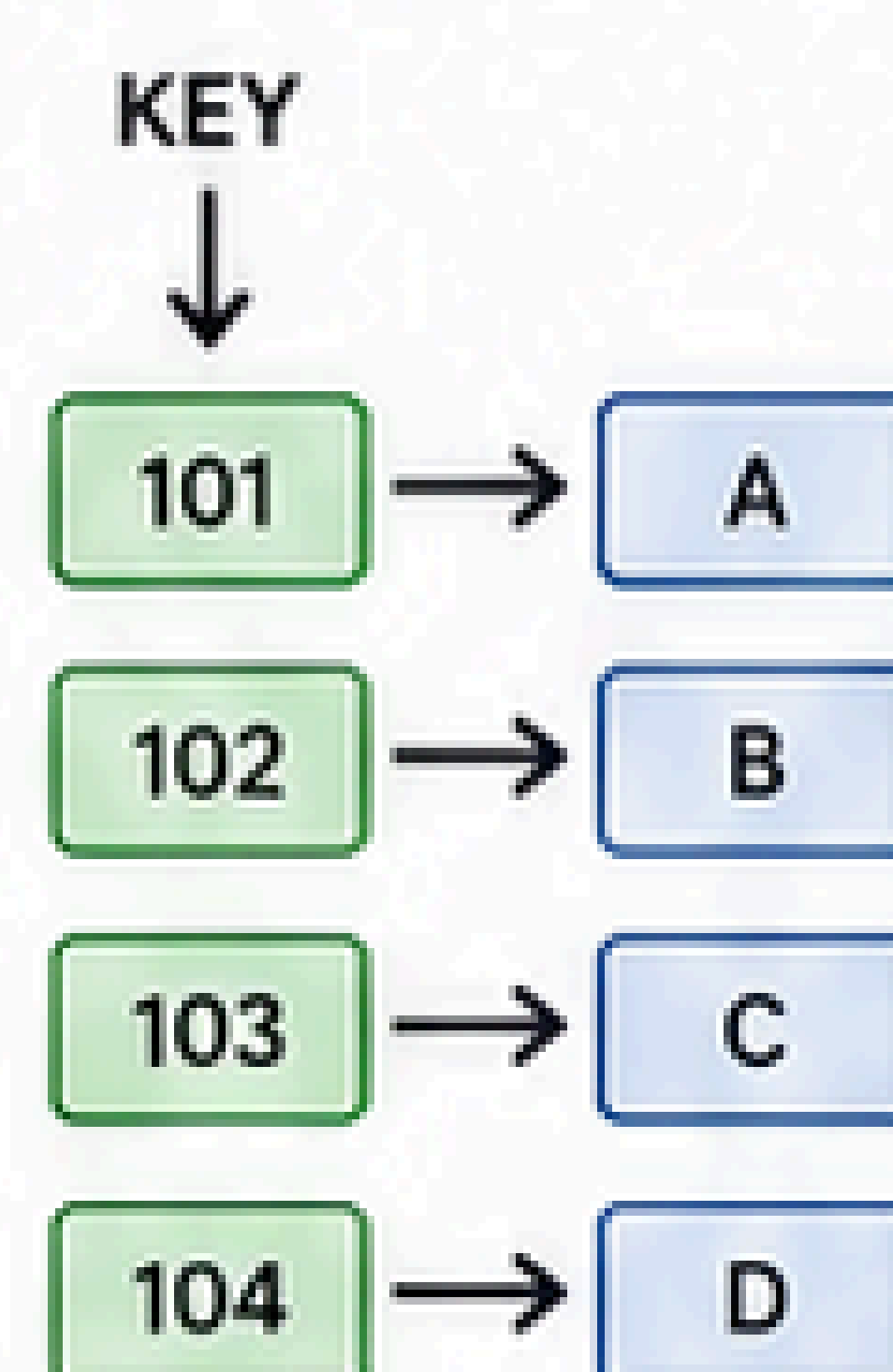
- queue
- insertion (add)
- deletion (remove)
- updation (peek)
- problem solving questions
- leetcode
- hacker rank
- codechef

## PART 7 – DP BASED QUESTIONS



Java  
dp based questions

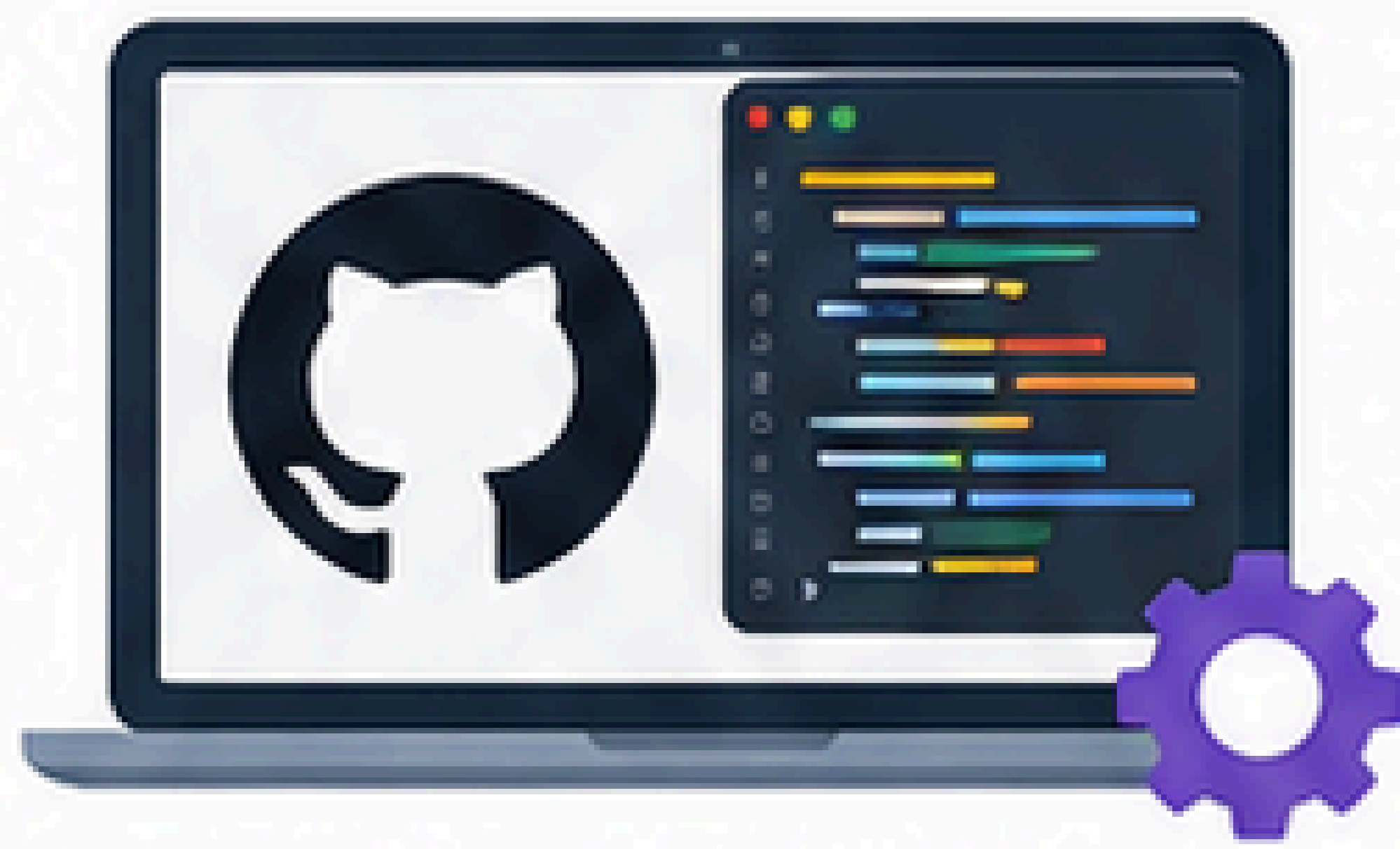
## PART 8 – HASHMAP



- hashmap
- insertion (put)
- deletion (remove)
- updation (replace)
- problem solving questions
- leetcode
- hacker rank
- codechef

# PHASE 6

## PART 1 – PROJECT



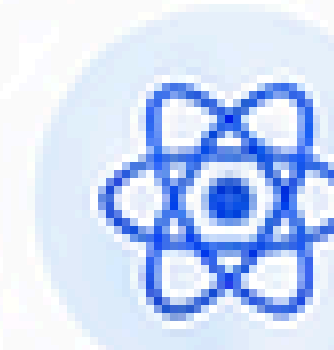
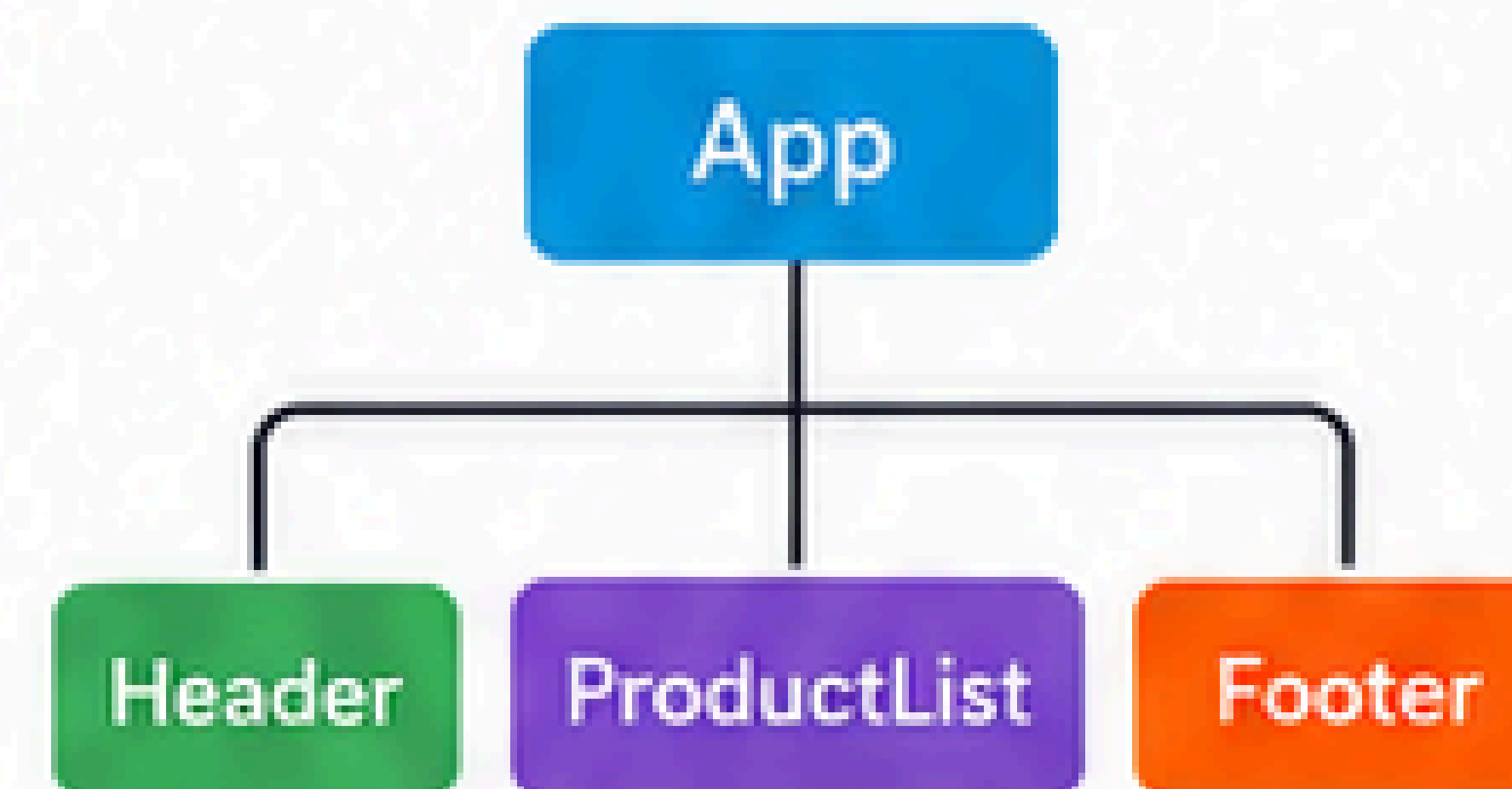
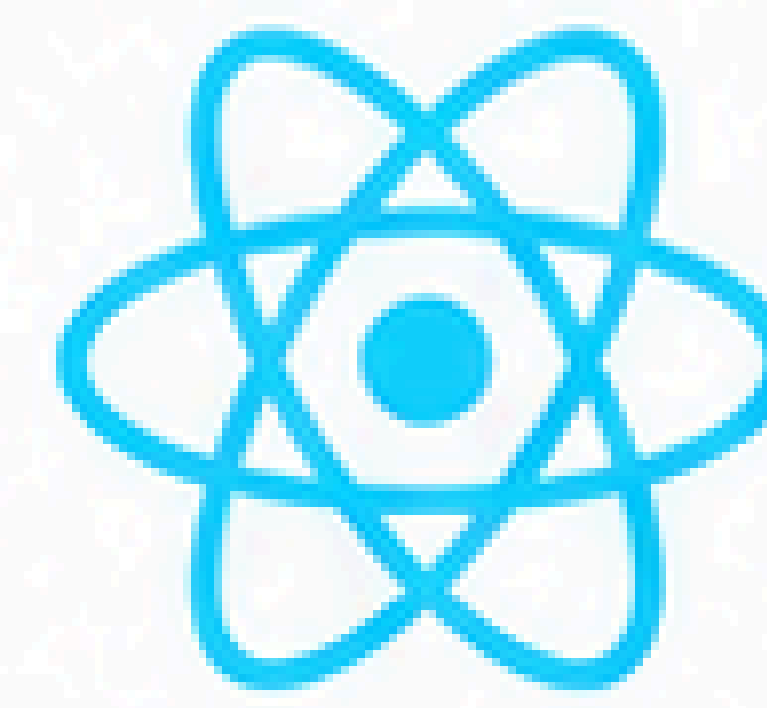
### Project

- Github API Fetching Project
- Product App

### PRODUCTS



## PART 2 – REACT BASICS



React  
Introduction to React.js



Components



Props

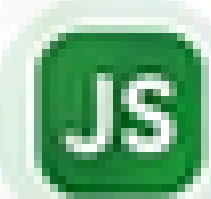


State

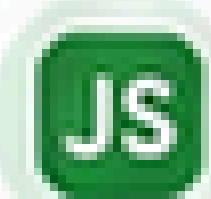


Handling Events

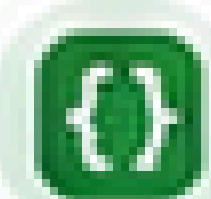
## PART 3 – JAVASCRIPT (FRONTEND BASICS)



Introduction to JavaScript



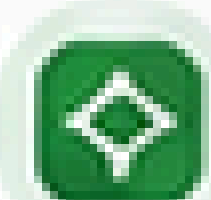
Variables and Data Types



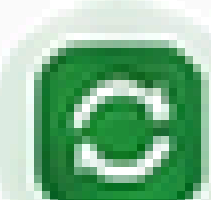
Let, Const, and Block Scope



Operators



Control Structures (if, switch)

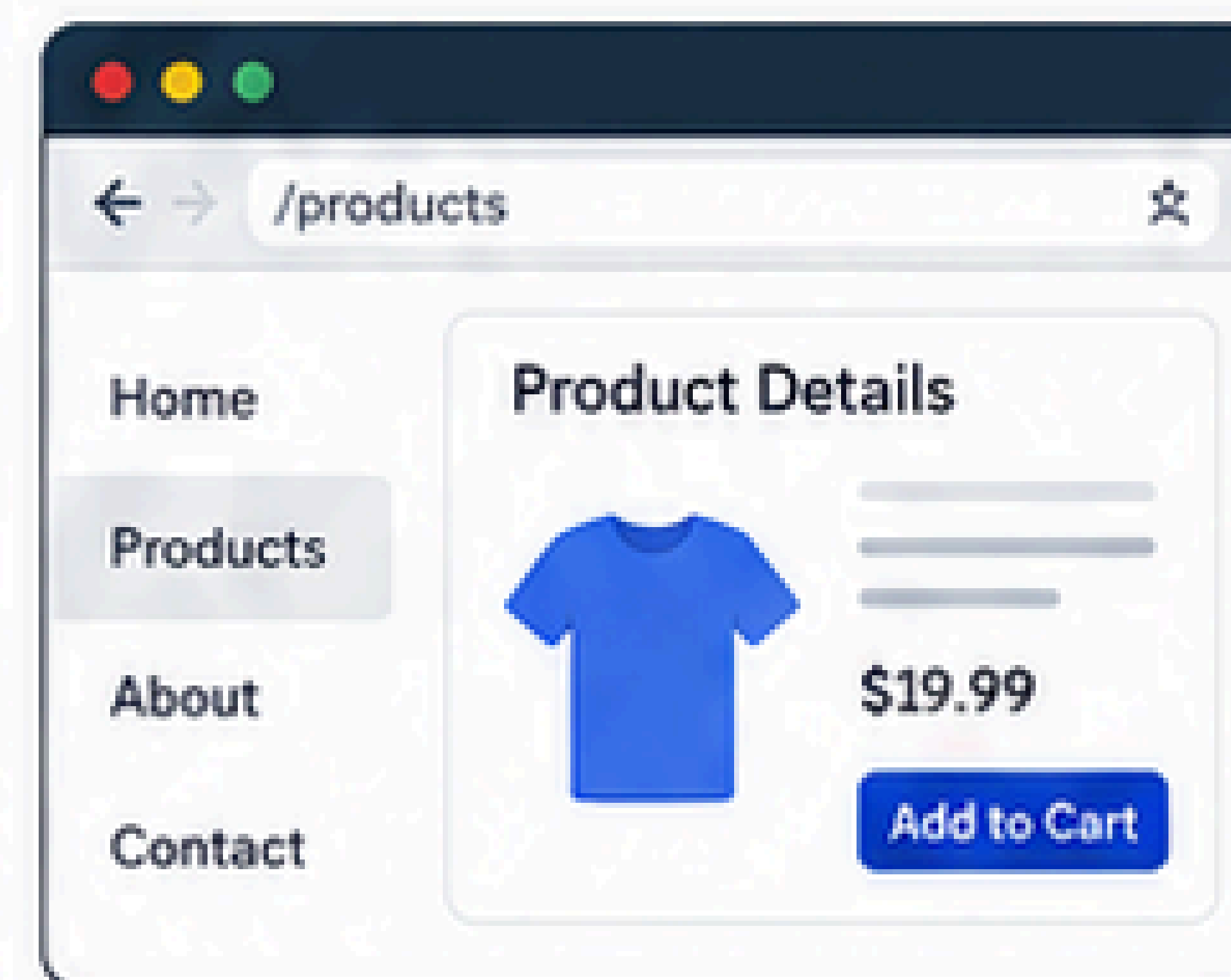


Loops (for, while)

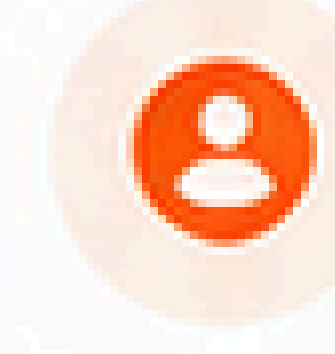


String and String Methods

## PART 4 – REACT ROUTER & STATE MANAGEMENT



React Router



Context Api



Two way data bindings

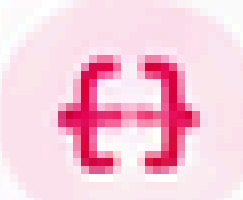


State Management

## PART 5 – JAVASCRIPT (ADVANCED)



Functions and Scope



Arrays and Objects



DOM Manipulation



Events and Event Handling

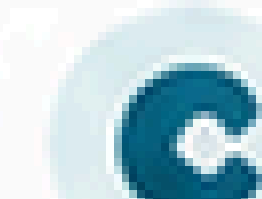
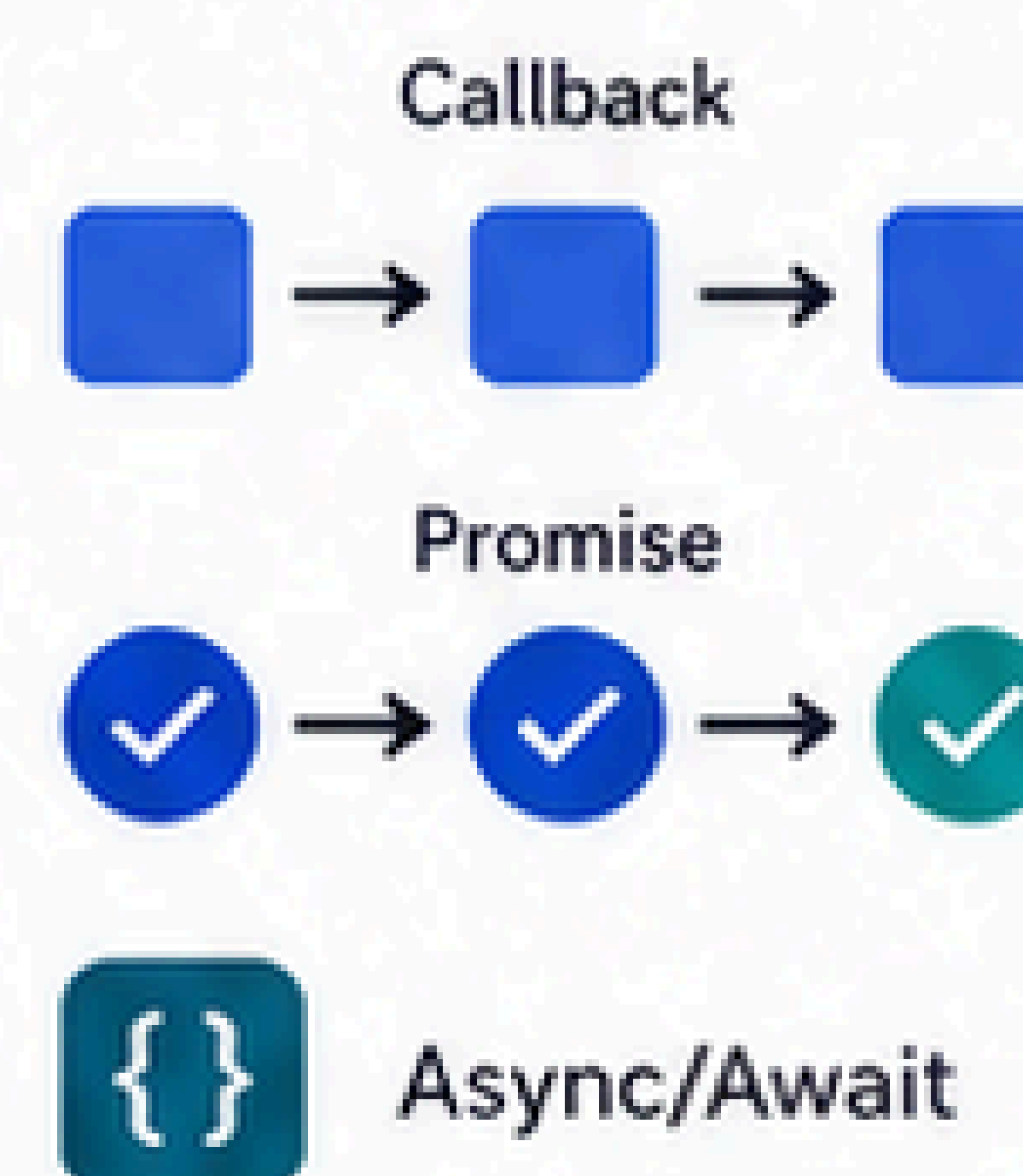


Destructuring



Spread and Rest Operators

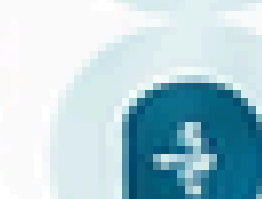
## PART 6 – ASYNC IN JAVASCRIPT



Callbacks



Promises



Async/Await

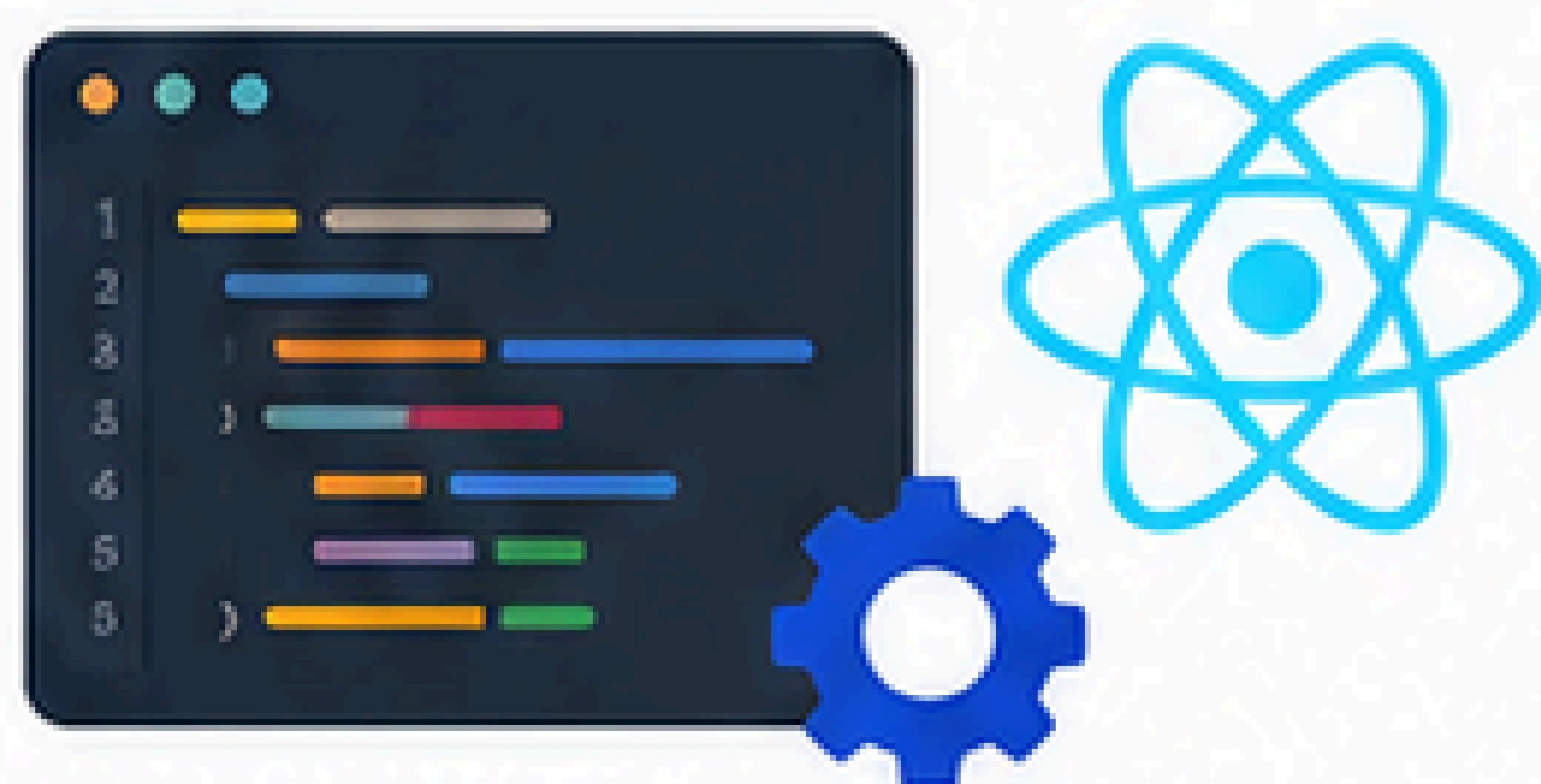


JSON and Fetch API



Error Handling

## PART 7 – ADVANCED REACT CONCEPTS



Advanced React Concepts and Styling



Hooks (useState, useEffect, useContext, useRef, etc.)



Lifecycle Methods



Styling (CSS Modules, Styled Components, Tailwind CSS, etc.)



Topic B (Additional Advanced Topics)

## PART 8 – INTEGRATION OVERVIEW



Build Full Product App



Fetch Data from Github API



Display Products

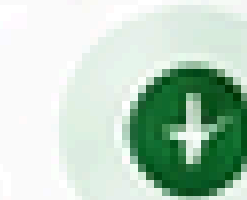
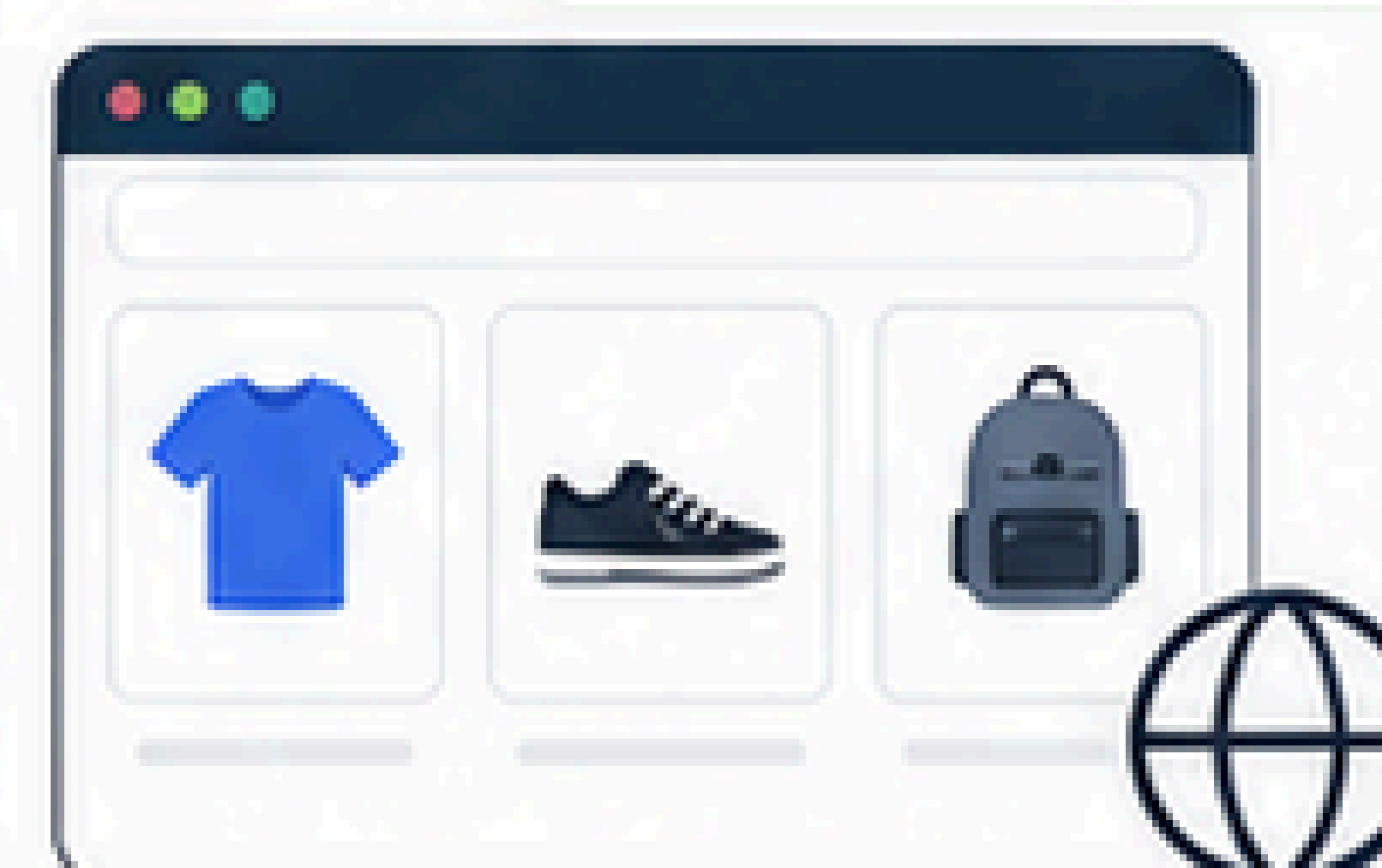


Routing and State Management

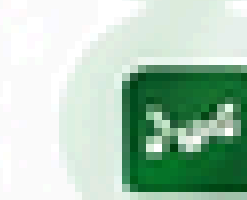


Handle Events and User Interaction

## PART 9 – BUILD & DEPLOY



Project Structuring



Build and Testing



Deployment (Netlify / Vercel)



Git and GitHub Workflow

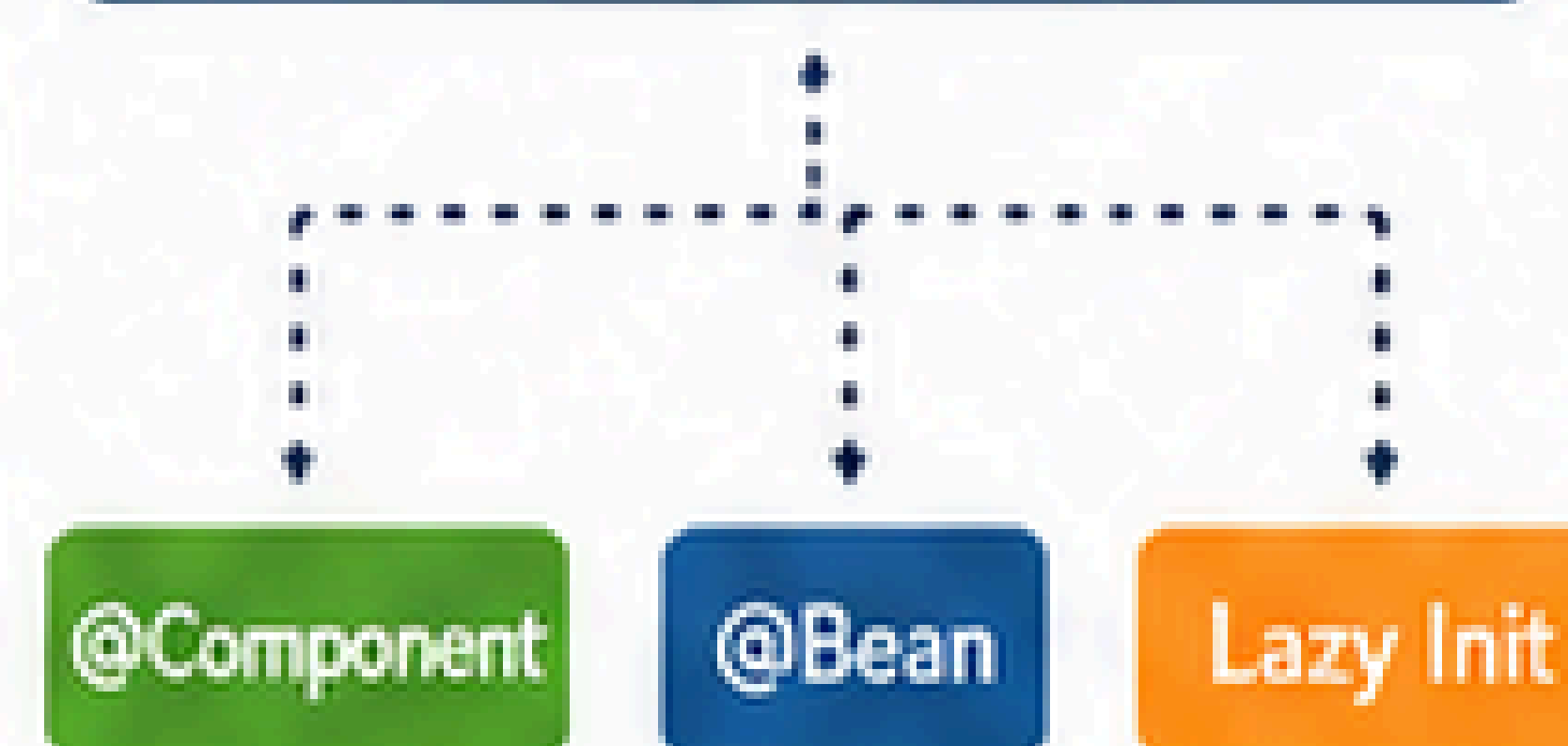
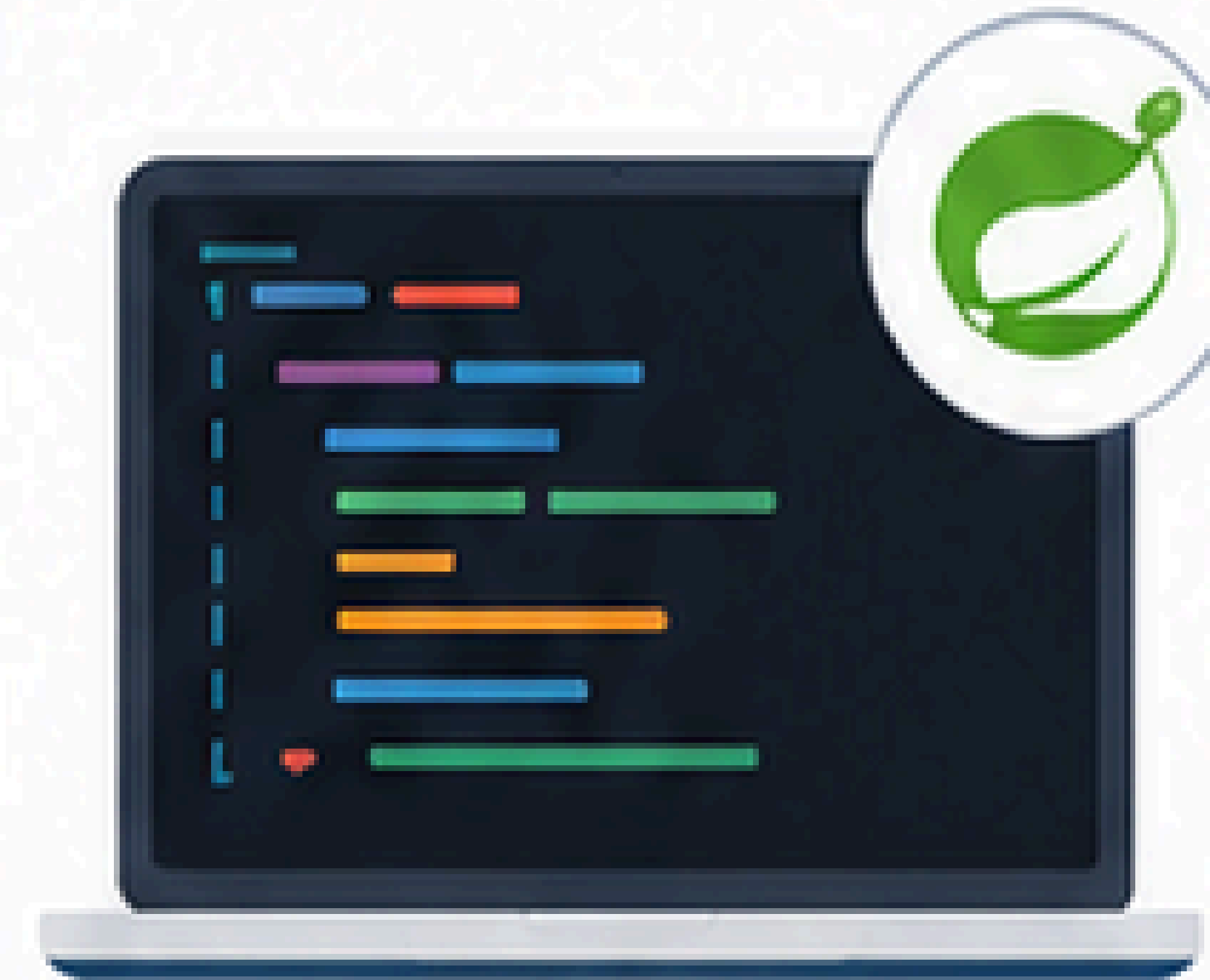
# PHASE 7

## PART 1 – SPRING CORE & DI BASICS



- Introduction to Spring
- Fundamentals Of Spring (Tight Coupling And Loose Coupling)
- Spring Beans and Spring Configuration
- Implementation as Auto Wiring and Spring IOC Container
- Primary and Qualifier
- + Implementation of all the above Topic in project

## PART 2 – SPRING CORE ADVANCED



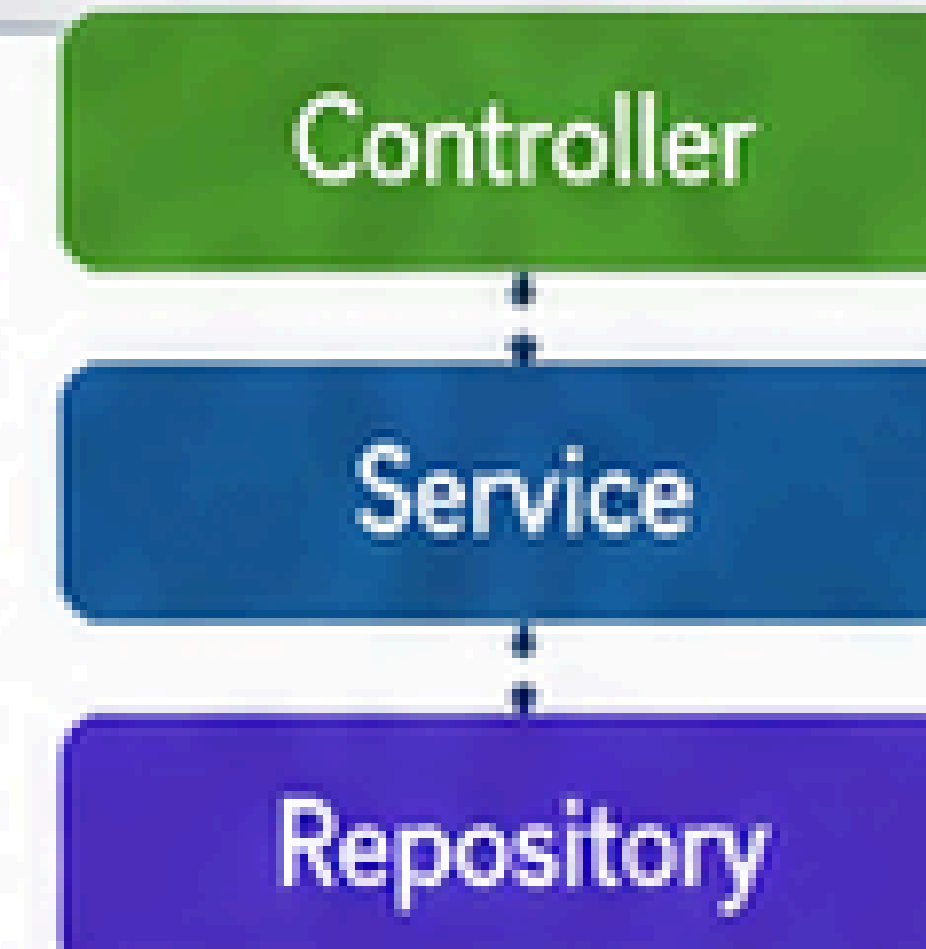
- Spring Core - Primary & Qualifier, Comparing @Component vs @Bean
- Exercise: Solution for Real World Java Spring Framework Example
- Exploring Lazy and Eager Initialization of Spring Framework Beans & its Differences
- Comparing Prototype vs Singleton
- PostConstruct and PreDestroy
- Exploring Jakarta CDI with Spring Framework and Java
- Exploring Java Spring XML Configuration
- Java Annotations vs XML Configuration

## PART 3 – SPRING BOOT & JPA BASICS



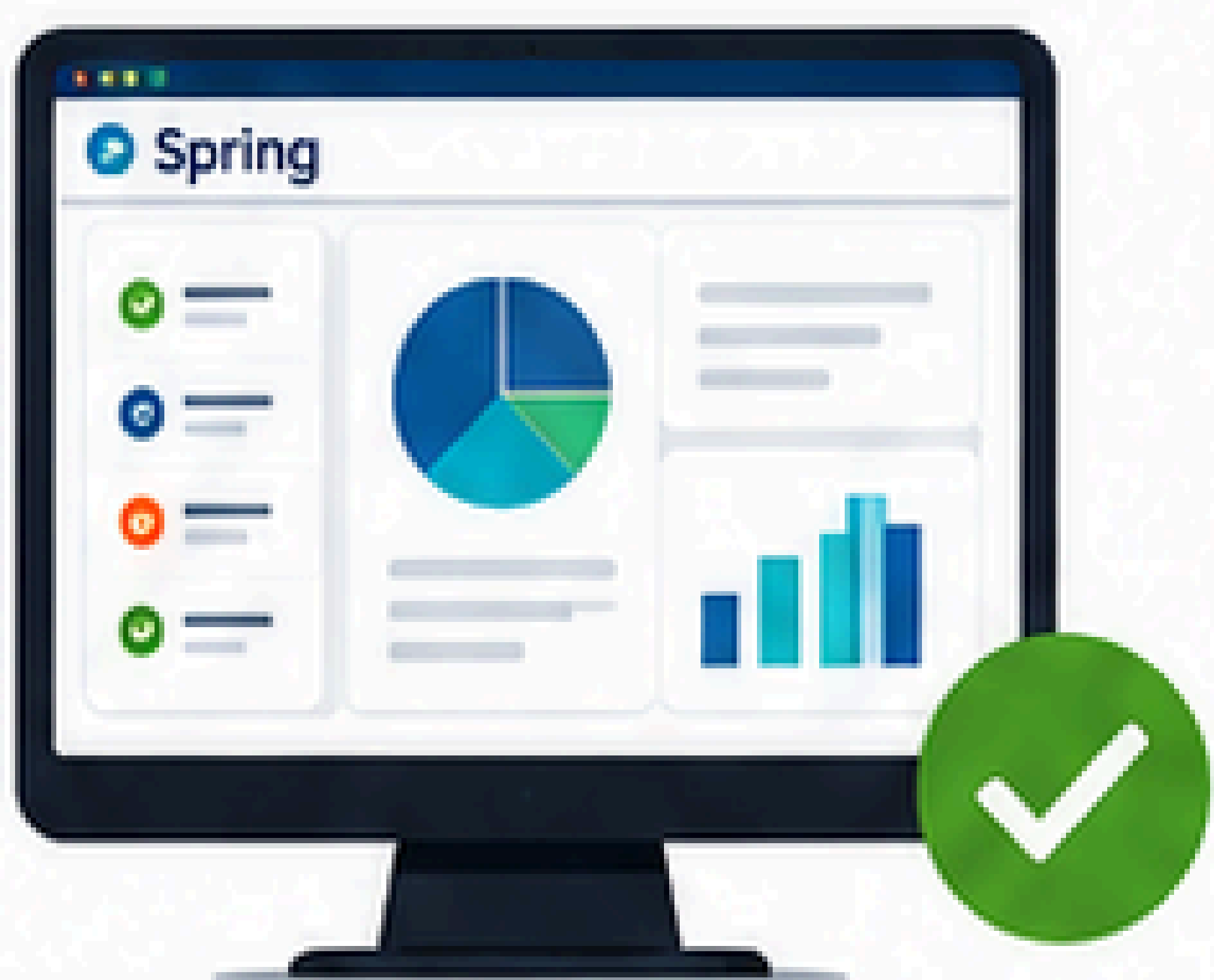
- Spring Boot introduction
- Fundamentals of Spring Boot
- JPA & Hibernate With Spring Boot
- JPA & Hibernate With Spring Boot - Spring JDBC
- JPA and EntityManager
- Spring Data JPA
- Features of Spring Data JPA

## PART 4 – SPRING BOOT WEB & CONTROLLERS



- Building Web App with Spring Boot
- Spring MVC Controller
- @ResponseBody
- @Controller
- Redirect to a JSP using Spring Boot
- Capturing Query Params using RequestParam

## PART 5 – SPRING BOOT PROJECT (WEB APPLICATION)

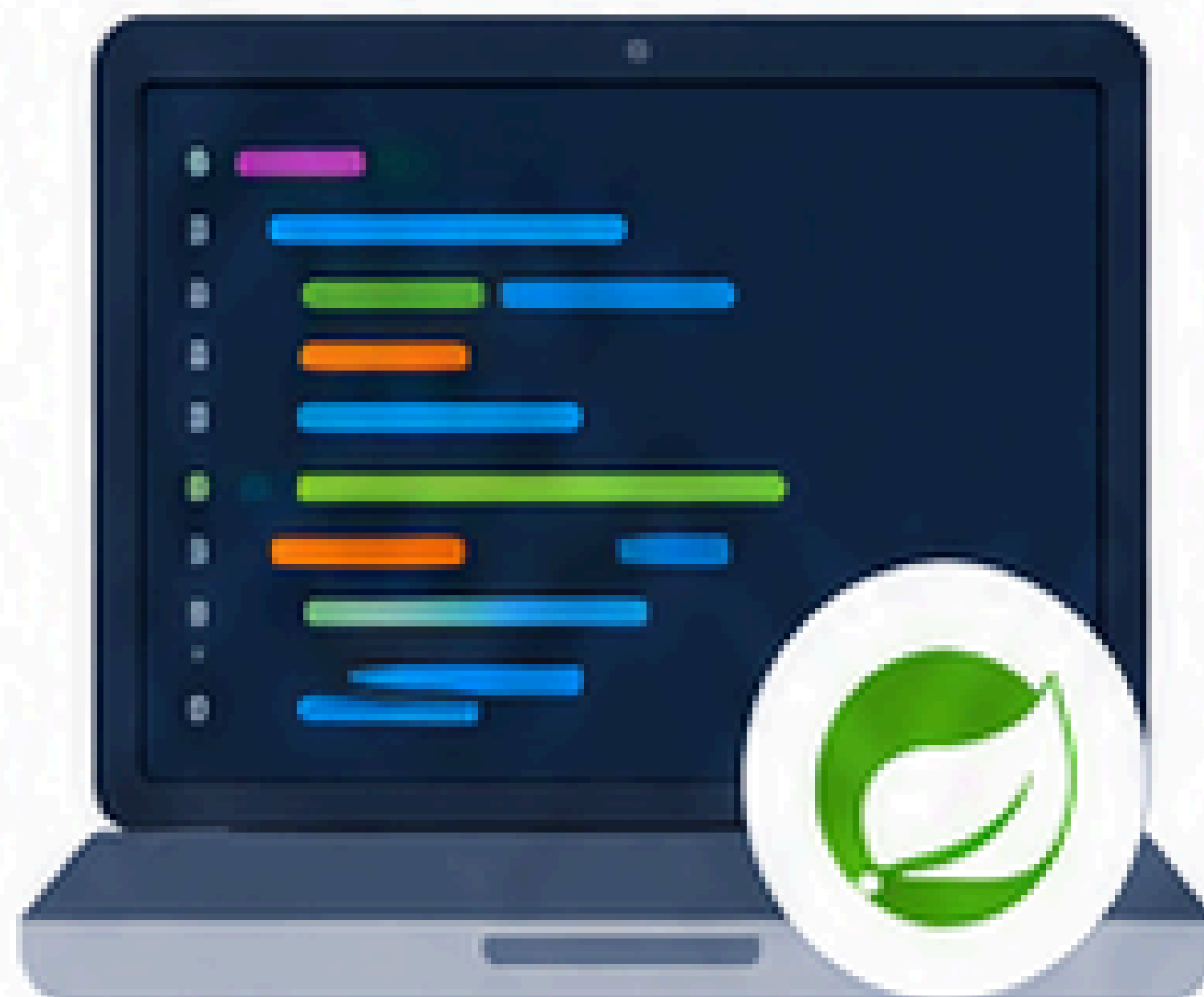


- Web Application using Spring-Spring Boot Framework
- CRUD application through project
- Verification and validation
- Connecting App to MYSQL database



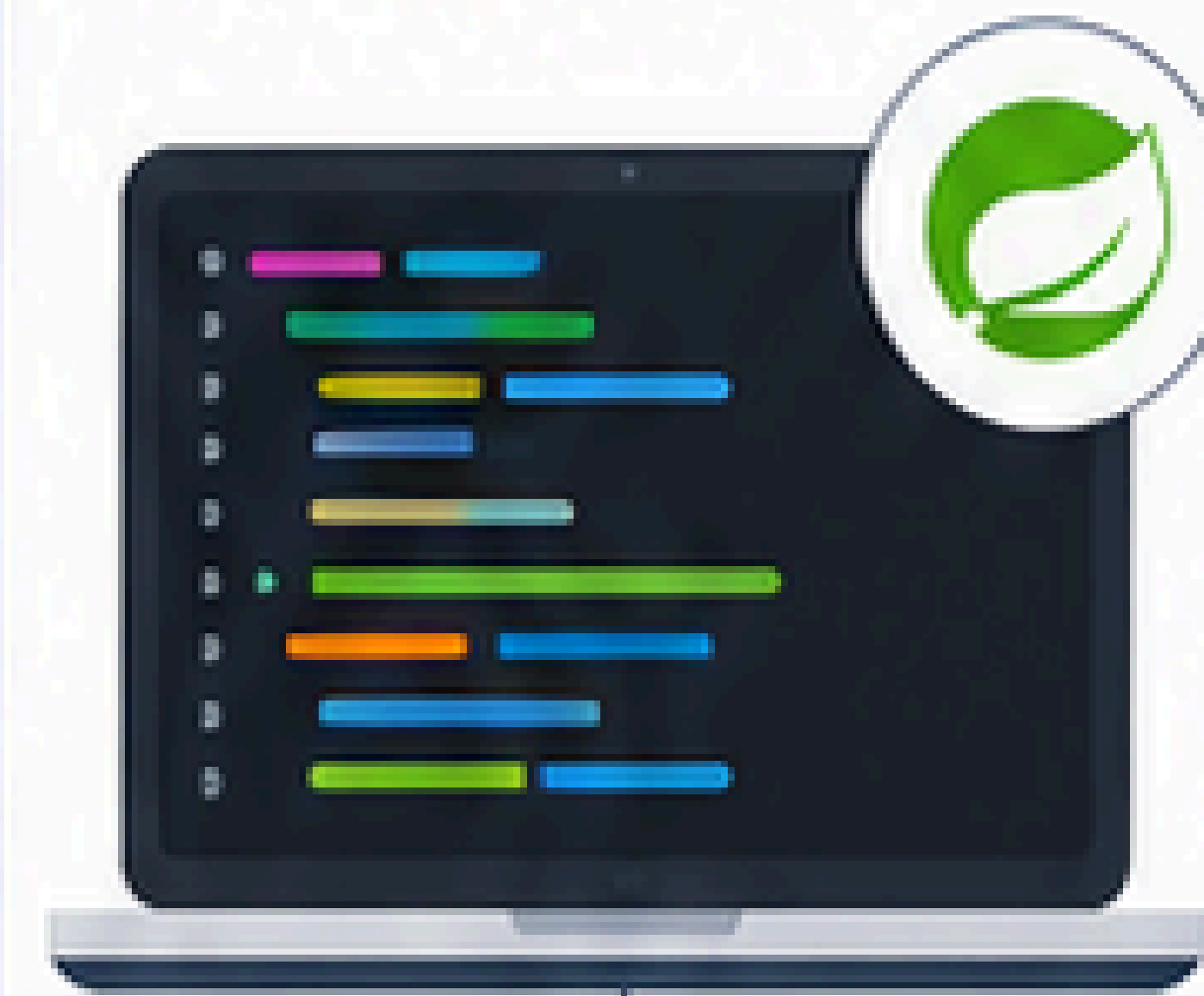
# PHASE 8

## PART 1 – SPRING FRAMEWORK FUNDAMENTALS



- ✓ Introduction to Spring
- ✓ Fundamentals Of Spring (Tight Coupling And Loose Coupling)
- ✓ Spring Beans and Spring Configuration
- ✓ Implementation as Auto Wiring and Spring IOC Container
- ✓ Primary and Qualifier
- ✓ + Implementation of all the above Topic in project

## PART 2 – SPRING CORE ADVANCED



- ✓ Spring Core - Primary & Qualifier, Comparing @Component vs @Bean
- ✓ Exercise: Solution for Real World Java Spring Framework Example
- ✓ Exploring Lazy and Eager Initialization of Spring Framework Beans & its Differences
- ✓ Comparing Prototype vs Singleton
- ✓ PostConstruct and PreDestroy
- ✓ Exploring Jakarta CDI with Spring Framework and Java
- ✓ Exploring Java Spring XML Configuration
- ✓ Java Annotations vs XML Configuration

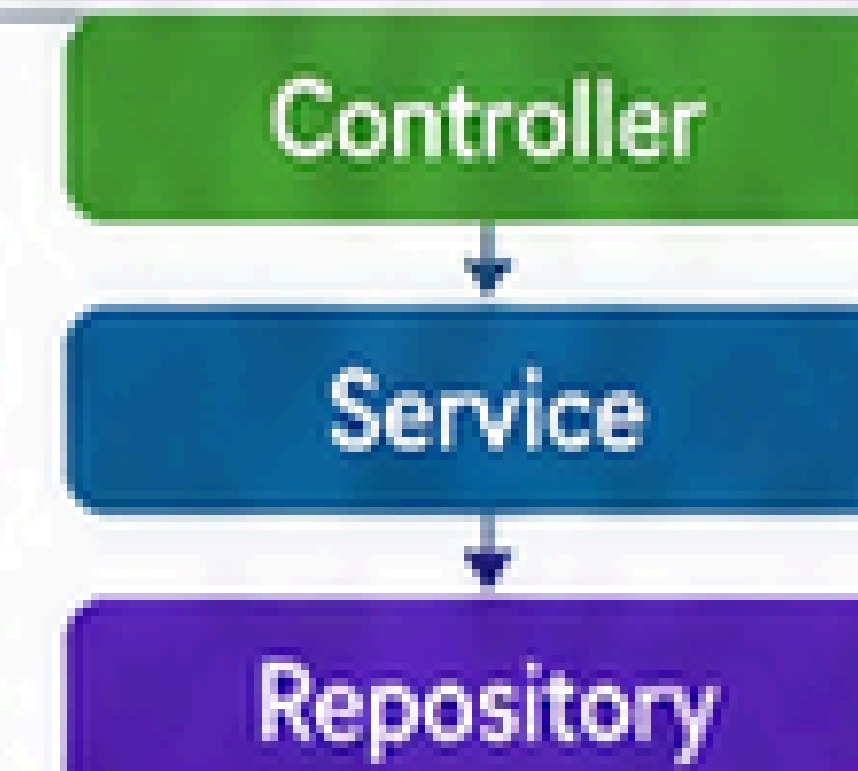
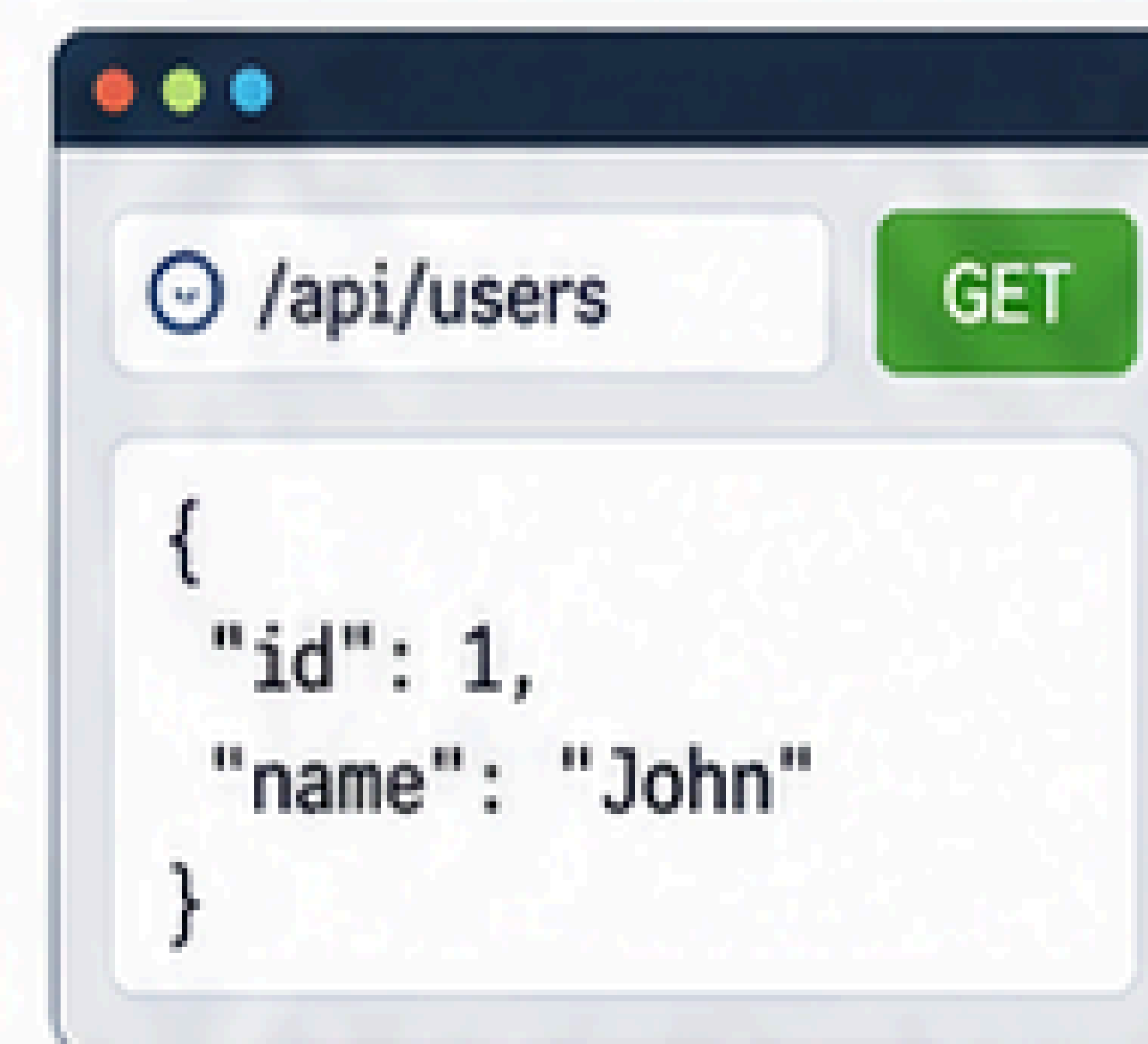
## PART 3 – SPRING BOOT & JPA BASICS



JPA /  
Hibernate

- ✓ Spring Boot introduction
- ✓ Fundamentals of Spring Boot
- ✓ JPA & Hibernate With Spring Boot
- ✓ JPA & Hibernate With Spring Boot - Spring JDBC
- ✓ JPA and EntityManager
- ✓ Spring Data JPA
- ✓ Features of Spring Data JPA

## PART 4 – SPRING BOOT WEB & CONTROLLERS



- ✓ Building Web App with Spring Boot
- ✓ Spring MVC Controller
- ✓ @ResponseBody
- ✓ @Controller
- ✓ Redirect to a JSP using Spring Boot
- ✓ Capturing Query Params using RequestParam

## PART 5 – SPRING BOOT PROJECT (WEB APPLICATION)



- ✓ Web Application using Spring-Spring Boot Framework
- ✓ CRUD application through project
- ✓ Verification and validation
- ✓ Connecting App to MYSQL database

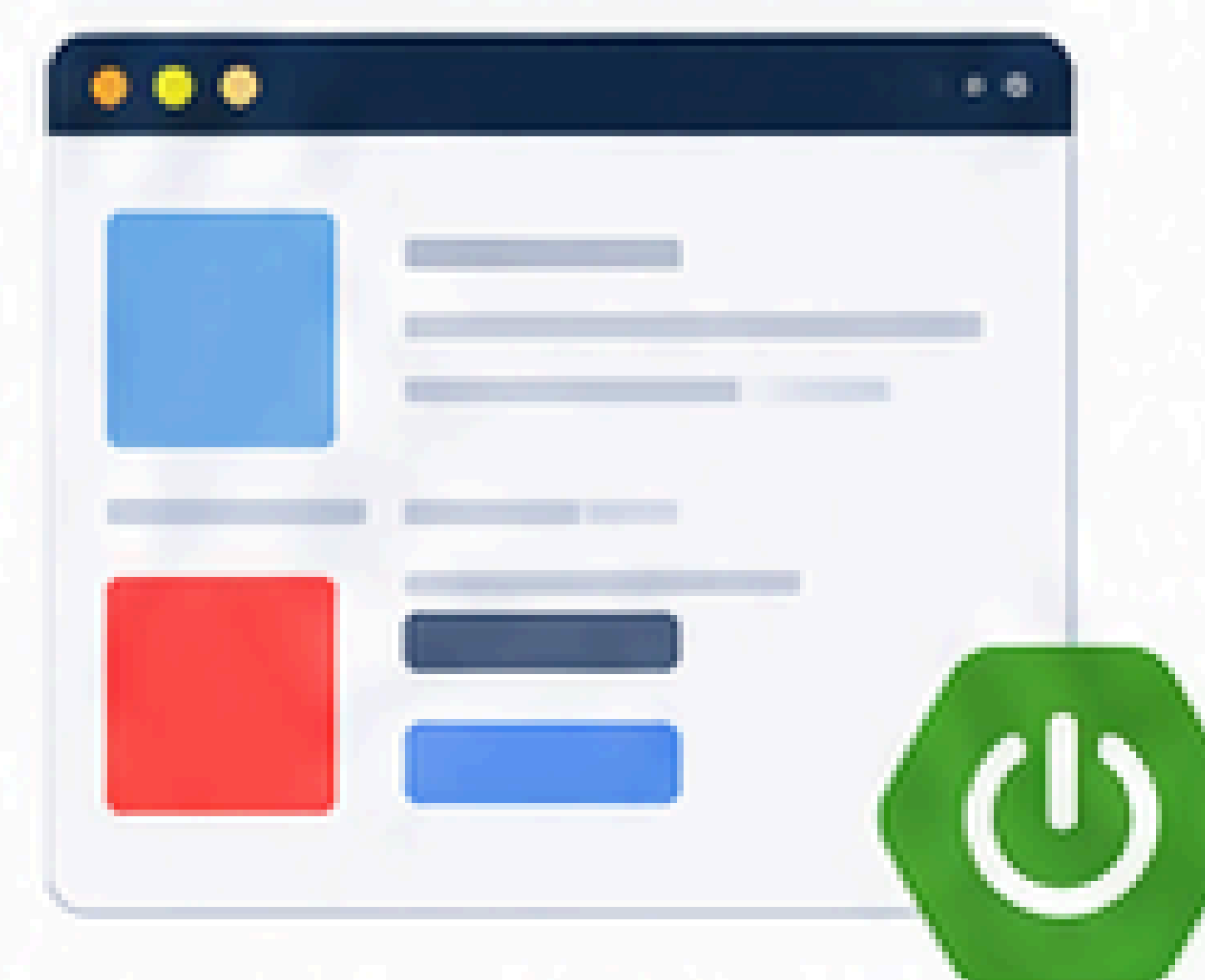


## PART 6 – ADVANCE JAVA WITH SPRING BOOT



- ✓ Spring Boot introduction , Fundamentals of Spring Boot , JPA & Hibernate With Spring
- ✓ - JPA & Hibernate With Spring Boot - Spring JDBC , JPA and EntityManager , Spring Data JPA , features of Spring Data JPA

## PART 7 – BUILDING WEB APP WITH SPRING BOOT



- ✓ Building Web App with Spring Boot
- ✓ Spring MVC Controller, @ResponseBody , @Controller ,Redirect to a JSP using Spring Boot ,
- ✓ Capturing QueryParams using RequestParam,

# PHASE 9

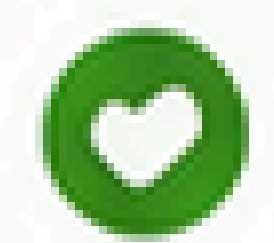
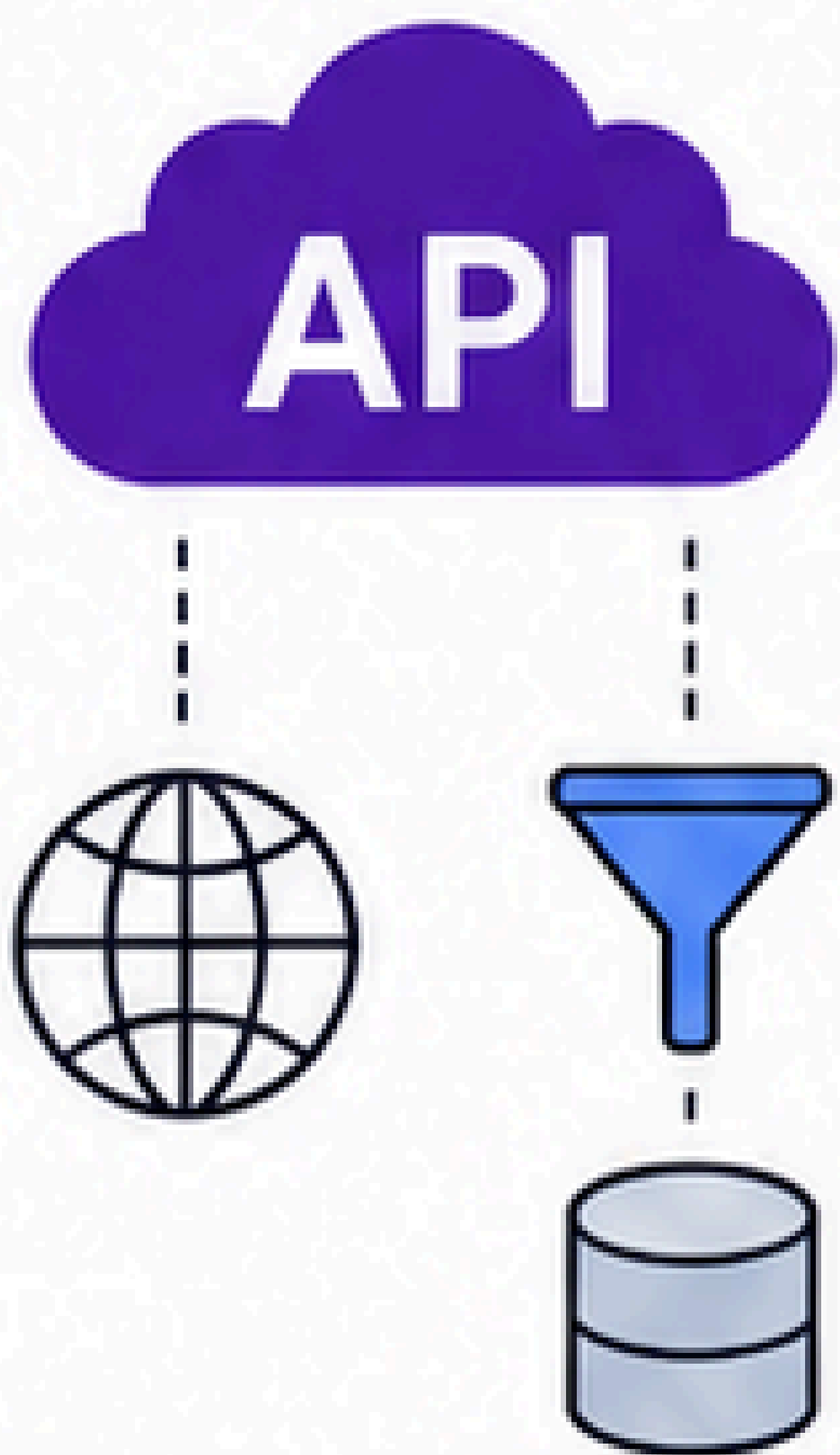
## PART 1 – FULL STACK ADVANCED PROJECT



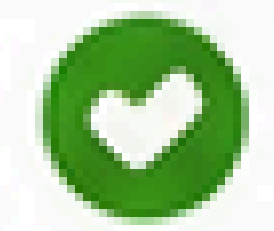
### Advance Project

Full Stack React & Spring Boot with JPA & Hibernate

## PART 2 – REST API ADVANCED TOPICS



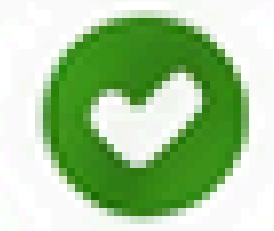
### Advance Project



Exploring Internationalization for REST API



Versioning REST API



Implementing Static Filtering



Implementing Dynamic Filtering



JPA and Hibernate Queries for REST API

## PART 3 – SPRING BOOT + REACT INTEGRATION



### Advance Project

Building Spring Boot web Application with React & Connecting Spring Boot REST API with React

## PART 4 – REST API PROJECTS (CORE CONCEPTS)



### Projects



REST API Introduction



Spring Boot Starters & Autoconfig



Path Variable



Designing the REST API

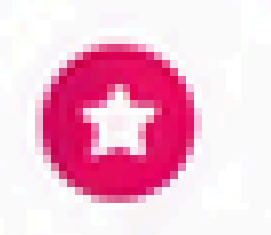
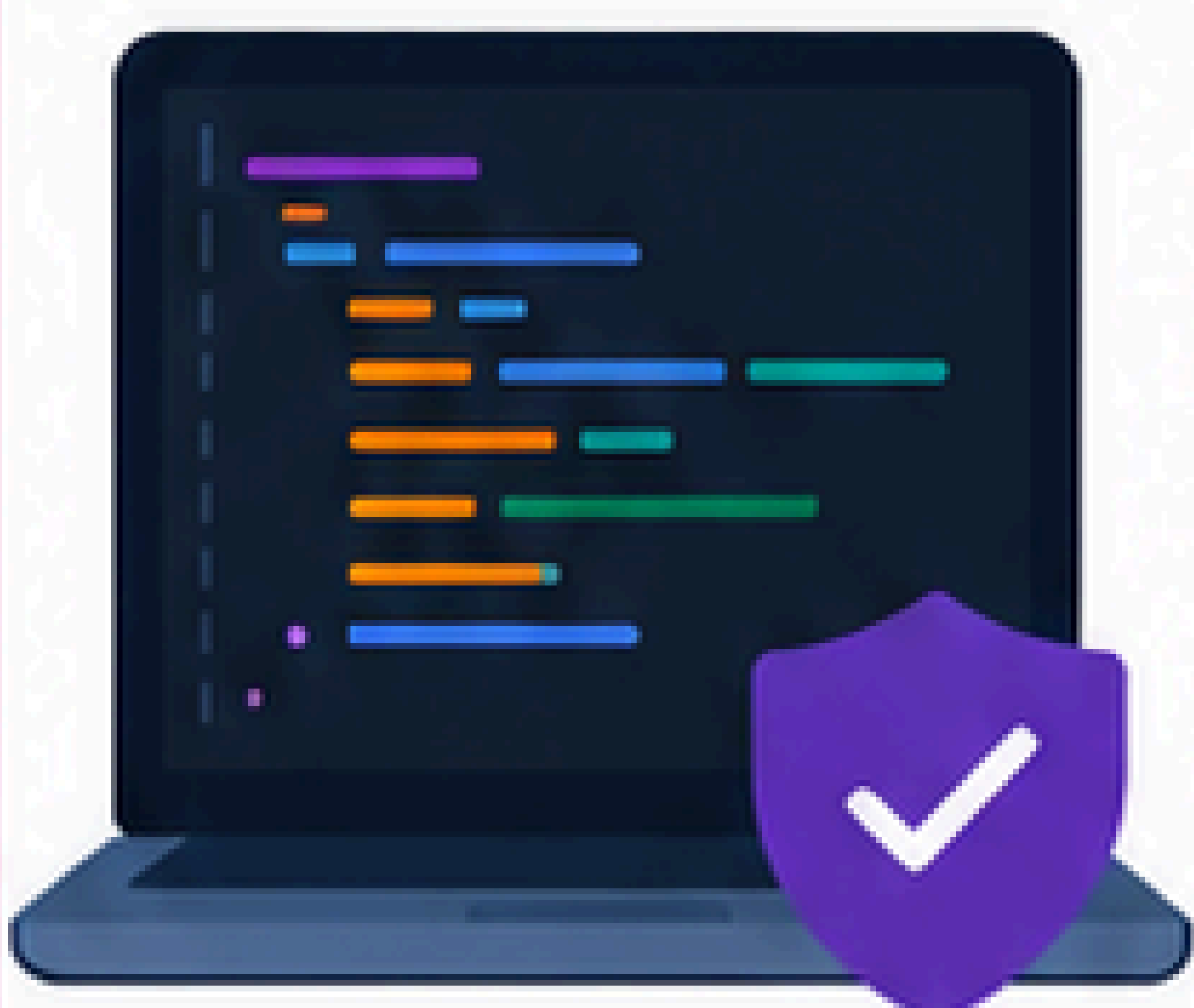


User Bean and UserDaoService

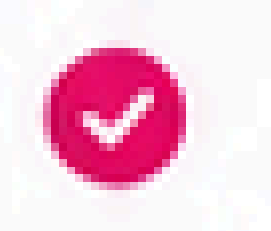


GET , POST & Exception Handling

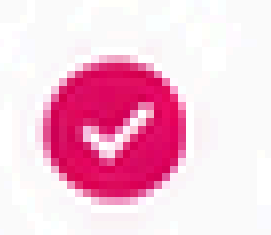
## PART 5 – ADVANCED REST API FEATURES



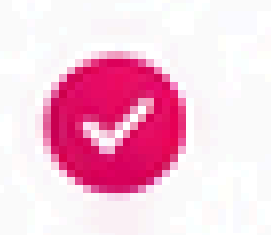
### Project



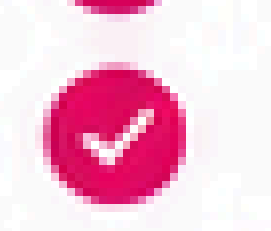
Implementing Generic Exception Handling for all Resources



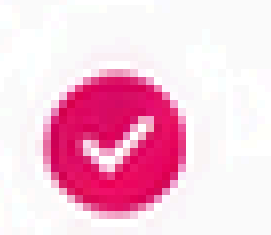
Implementing DELETE Method to delete a User Resource



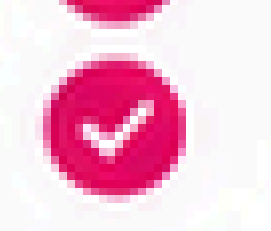
Implementing Validations



Overview of Advanced REST API Features



Open API Specification and Swagger



Configuring Auto Generation of Swagger Documentation

## PART 6 – MAJOR PROJECTS



### Project 1

Social Media II



### Project 2

E-commerce II



### Project 3

E-Wallet App like Paytm II



### Project 4

Movie Ticket Booking System

# JOIN OUR COMMUNITY:



**Telegram  
Channel**

<https://t.me/REGexSoftware>



**YouTube  
Channel**

[https://www.youtube.com/  
@REGexSoftware](https://www.youtube.com/@REGexSoftware)



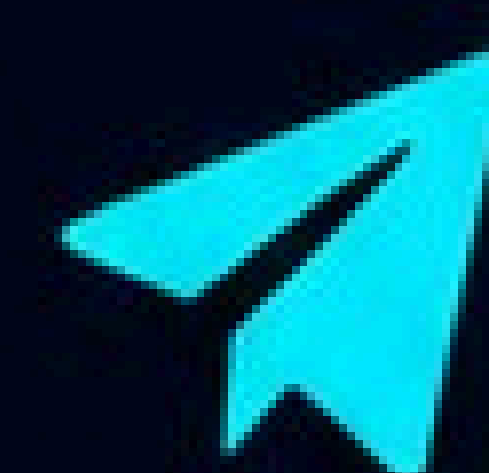
**Instagram  
Profile**

[https://www.instagram.com/  
regexsoftware/?hl=en](https://www.instagram.com/regexsoftware/?hl=en)



**LinkedIn  
Company Page**

[https://www.linkedin.com/  
company/regexsoftware/  
posts/?feedView=all](https://www.linkedin.com/company/regexsoftware/posts/?feedView=all)



**JOIN TELEGRAM**



**JOIN YOUTUBE**



**JOIN INSTAGRAM**



**JOIN LINKEDIN**