

Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	7
Title:	Design using Object Oriented approach with emphasis on Cohesion and Coupling
Date of Performance:	
Roll No:	9567, 9552, 9623
Team Members:	Shruti Patil, Mrunal Kotambkar, Dhruv Mayekar

Rubrics for Evaluation:

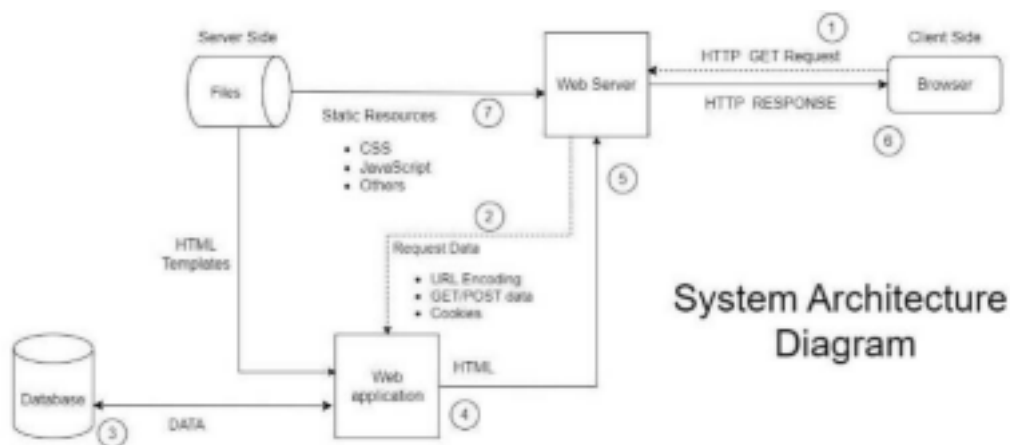
Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01(rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

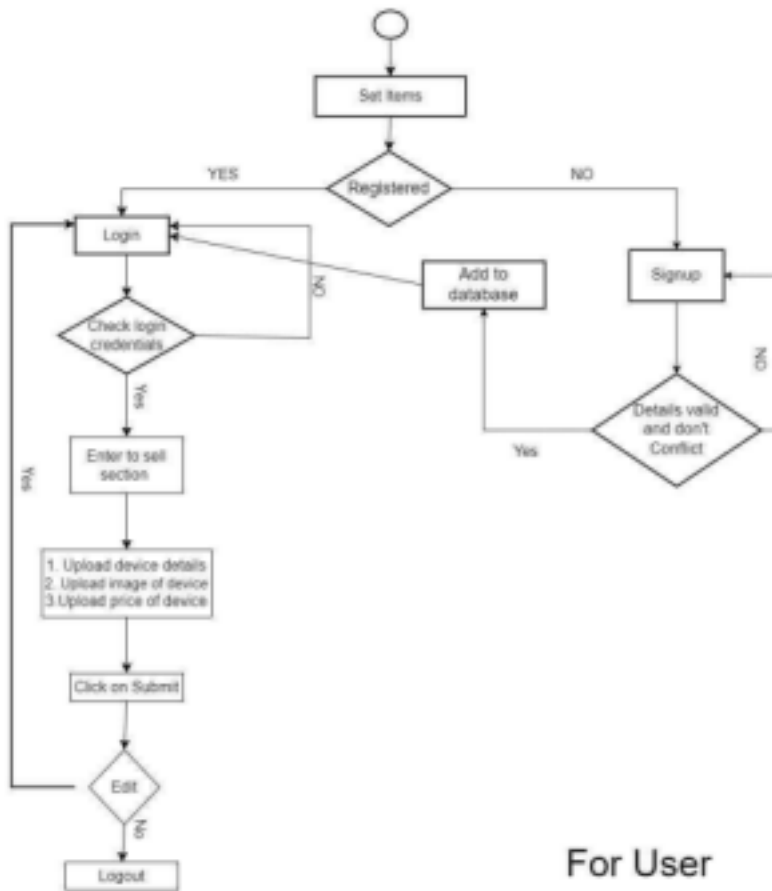
Signature of the Teacher:

Department of Computer Engineering

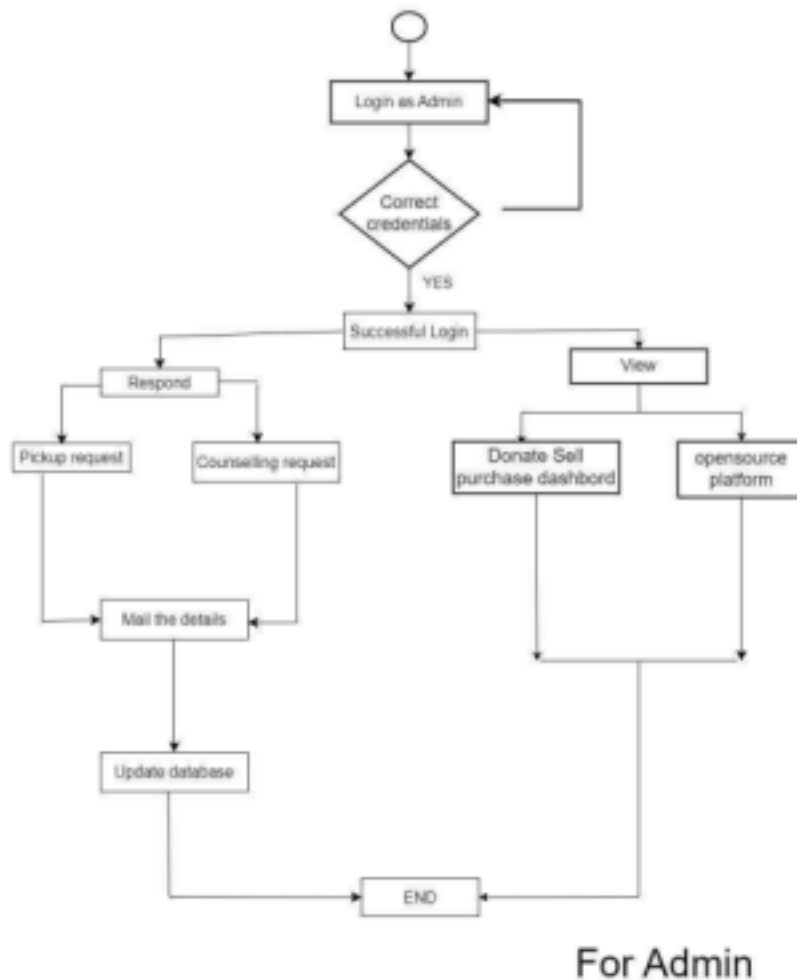
Academic Term: First Term 2022-23

Class: T.E /Computer Sem – V / Software Engineering Signature of the Teacher:





For User



POSTLAB:

- a) Analyse a given software design and assess the level of cohesion and coupling, identifying potential areas for improvement:

The software design of the "Samachar website demonstrates reasonably good levels of cohesion and coupling Cohesion is apparent in well-defined components with distinct functions, while loose coupling allows for flexibility and minimal interdependence between components. To improve the design, the website can benefit from further enhancing cohesion by minimizing overlapping functions and ensuring each component has a single, dar responsibility. Additionally, refining interfaces and interactions between components, focusing on modularity, and conducting regular testing and refactoring efforts are recommended for ongoing software quality and maintainability.

- b) Apply Object-Oriented principles, such as encapsulation and inheritance, to design a class hierarchy for a specific problem domain. In the "Vehicle" domain, we establish a class hierarchy using Object-Oriented principles: 1. Vehicle (Base Class) -Properties: make, model, year - Methods: start, stop, accelerate, brake 2 Car (Inberts from Vehide) -Additional Properties: numDoors, fuelType -Additional Methods: lockDoors, unlockDoors 3. Motorcycle (Inherits from Vehide): -Additional Properties: hasHelmet Storage -Additional Methods: putOnHelmet takeOffHelmet 4. Truck (Inhents from Vehide) -Additional Properties: cargoCapacity -Additional Methods: loadCargo, unload Cargo This hierarchy exemplifies encapsulation, where properties and methods are

contained within each class, and inheritance, which allows specialized classes to inherit properties and methods from the base class, promoting code reusability and structure.

c) Evaluate the impact of cohesion and coupling on software maintenance, extensibility, and reusability in a real-world project scenario. In a real-world project, cohesion and coupling have significant effects:

- Software Maintenance: High cohesion simplifies changes, and low coupling reduces unintended impacts during maintenance.
- Software Extensibility: High cohesion and low coupling ease the addition of new features and components.
- Software Reusability: Well-structured, cohesive, and loosely coupled code is more reusable in various contexts.

In practice, striking a balance between cohesion and coupling is crucial for business agility, cost savings, team collaboration, and quality assurance in long-term projects.