K. J. Somaiya College of Engineering, Mumbai-77

Batch: A1	**Roll No.: 16010121051**
Experiment / assignment / tutorial No 4	

TITLE : Software architecture of B Tech. ProjectArchitecture in two Known different styles

AIM : To understand different types of architectures.

Expected OUTCOME of Experiment:

CO 1. Design the architecture of software systems in various architectural styles.

Books/ Journals/ Websites referred:

1 “Software Architecture, Richard N Taylor etl, Wiley

www.isr.uci.edu/projects/archstudio/setup-easy.html

Aim: To Design the architecture of software systems in various architectural styles.

Title of B.E Project: Fund Trail Analysis Tool

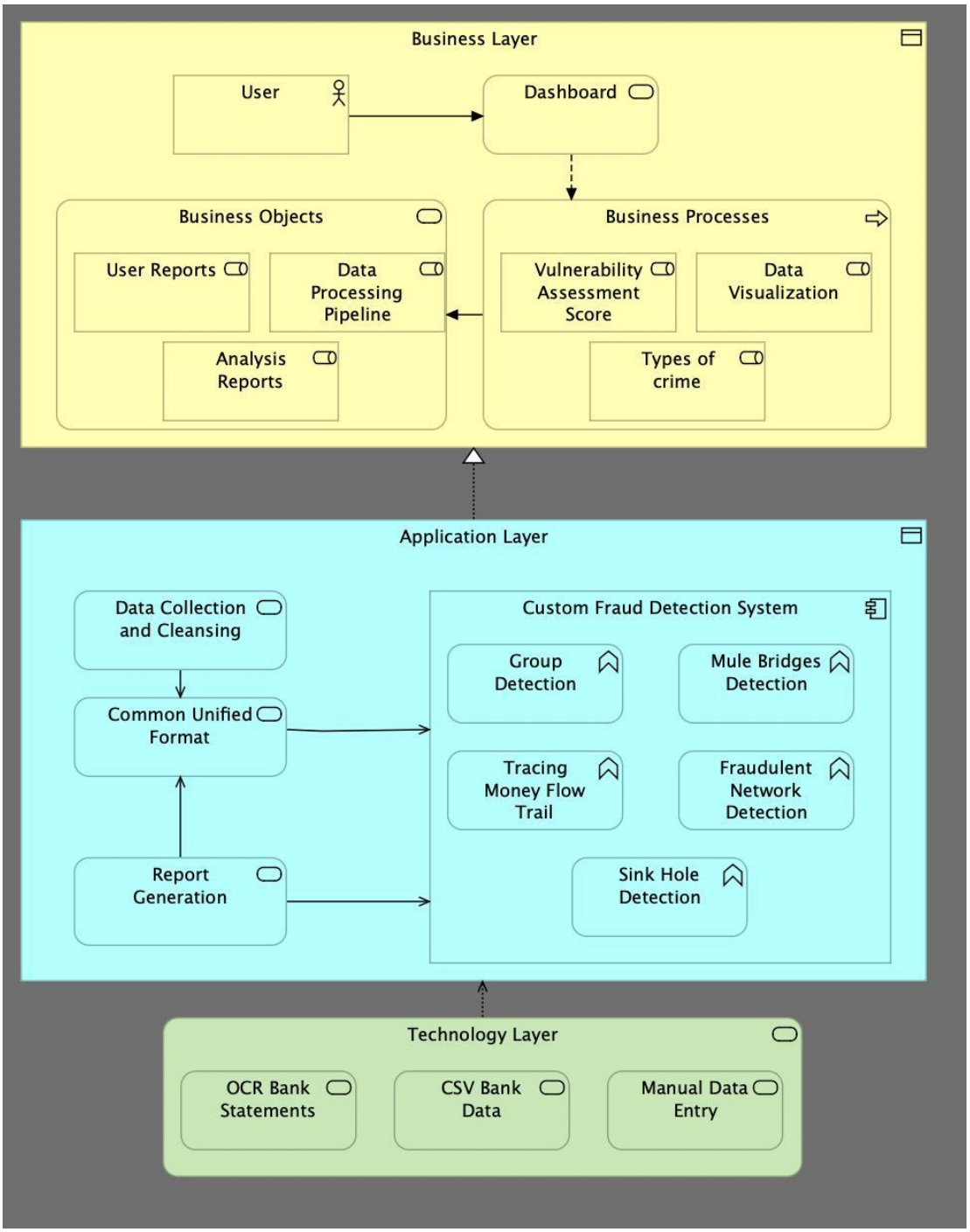
Abstract of B.E Project:

The "Fund Trail Analysis Tool" is designed to combat financial fraud and illicit transactions by leveraging advanced data analytics and graph-based algorithms. Over the years, India has suffered significant economic losses, amounting to approximately Rs 20 lakh crore from frauds between 1948 and 2008. This tool aims to be a pivotal solution in addressing these challenges by providing robust support for intergovernmental tax bodies and financial institutions in identifying complex money trails across various jurisdictions.

Key features of the tool include custom fraud detection algorithms and money trail flow analysis, which enhance the detection of suspicious transactions such as money laundering and embezzlement. The tool processes data from OCR-scanned bank statements and CSV files to generate comprehensive reports. Additionally, it employs balance sheet analysis to detect potential financial sinkholes, a critical aspect of proactive

financial risk management.
By integrating these functionalities, the tool can significantly reduce illicit financial activities by enabling early detection and tracking of suspicious transactions, thereby supporting regulatory bodies and enhancing global financial transparency.

Architecture of .B.E Project:



Explanation of Components and Connectors:

Here is an explanation of the components and connectors, along with the architectural styles used:

Components:

1. **Business Layer:**
    - **User:** Represents the end user interacting with the system.
    - **Dashboard:** The primary interface for users to visualize and manage reports, scores, and data related to vulnerabilities.
    - **Business Objects:**
    - **User Reports:** Displays reports created or generated for users.
    - **Data Processing Pipeline:** Processes data from various sources and feeds it into the necessary systems for analysis and reporting.
    - **Analysis Reports:** Compiles reports on data insights after processing.
    - **Business Processes:**
    - **Vulnerability Assessment Score:** The system calculates vulnerability scores based on data inputs.
    - **Data Visualization:** Provides visual insights into processed data.
    - **Types of Crime:** Categories or types of crimes identified through the system.
2. **Application Layer:**
    - **Data Collection and Cleansing:** Gathers data from different sources and cleans it for processing.
    - **Common Unified Format:** Standardizes the data into a unified format for further processing.
    - **Custom Fraud Detection System:** Detects various types of fraud, including:
        - **Group Detection:** Identifies groups involved in fraudulent activities.
        - **Tracing Money Flow Trail:** Tracks the flow of money for potential fraud detection.
        - **Mule Bridges Detection:** Detects intermediaries (mules) used in fraud schemes.
        - **Fraudulent Network Detection:** Uncovers networks involved in fraudulent transactions.
        - **Sink Hole Detection:** Detects financial sinkholes (fraudulent transaction patterns).
    - **Report Generation:** Produces reports based on the processed and analysed data.
3. **Technology Layer:**
    - **OCR Bank Statements:** Optical Character Recognition is used to convert bank statements into digital data.
    - **CSV Bank Data:** Data extracted from CSV files related to banking transactions.
    - **Manual Data Entry:** Allows for manual input of data into the system.

Connectors:

1. **User Interaction with the Dashboard:** The user accesses the dashboard to view reports and data visualizations.
2. **Dashboard to Business Objects/Processes:** The dashboard displays data processed by the pipeline and business objects like reports.
3. **Data Flow between Layers:**
    - Data is collected from the technology layer (bank statements, CSV data, or manual entry), cleansed, and converted into a unified format in the application layer.
    - This data is then processed through the custom fraud detection system, analyzed, and visualized in the business layer.
4. **Report Generation and Interaction:** Once the data is processed, reports are generated and available for the user to review via the dashboard.

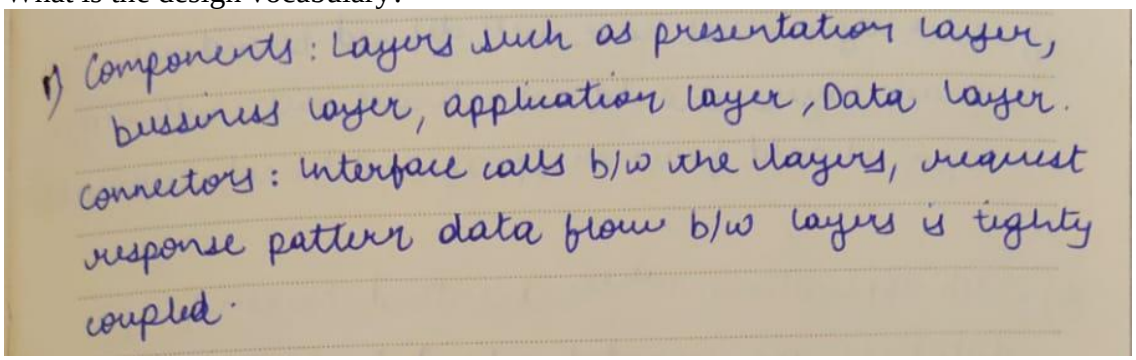
Architecture Styles Used:

1. **Layered Architecture:**
    - **Justification:** The system is clearly divided into three distinct layers: Business, Application, and Technology. Each layer has defined responsibilities, ensuring separation of concerns and modularity.
2. **Pipeline Architecture:**
    - **Justification:** The data flows from one stage to another, especially from the technology layer (data collection) to the business layer (data processing and visualization), following a sequential process.
3. **Client-Server Architecture:**
    - **Justification:** The system follows a client-server interaction where the user (client) interacts with the business layer (server) via the dashboard to access reports and insights.

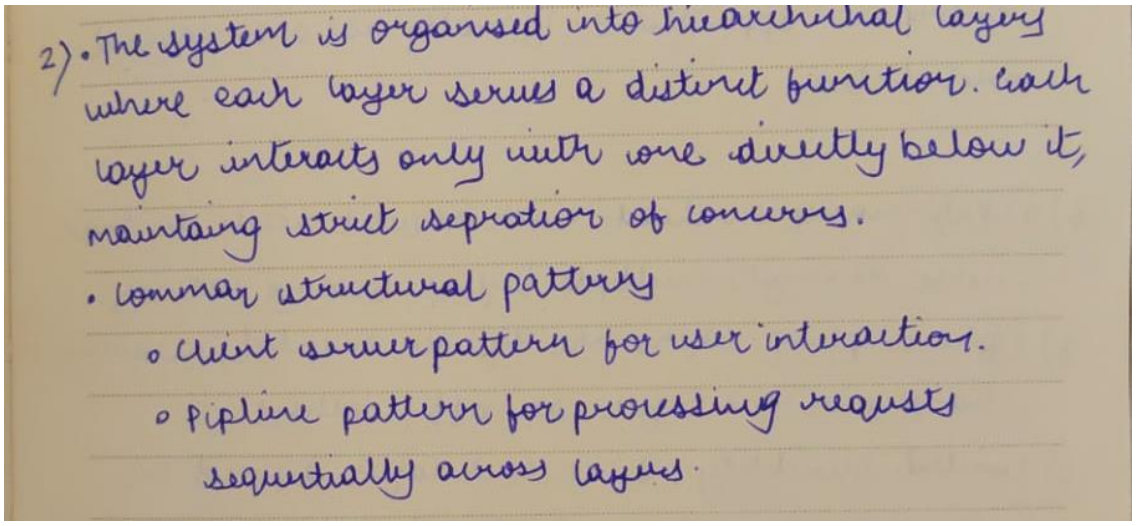
Post Lab Descriptive Questions

1. Design and Compare the architecture of BE project in 2 different architecture style w.r.t Style Analysis Dimensions

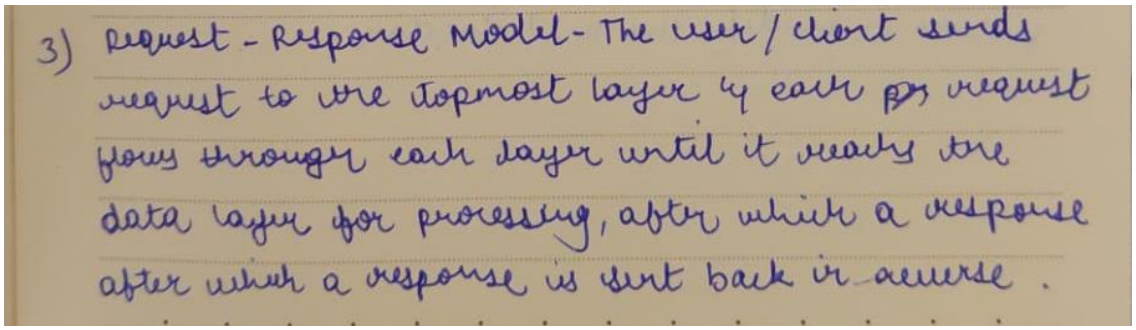
1. What is the design vocabulary?


1) Components : Layers such as presentation layer, bussiness layer, application layer, Data layer.
Connectors : Interface calls b/w the layers, request response pattern data flow b/w layers is tightly coupled.

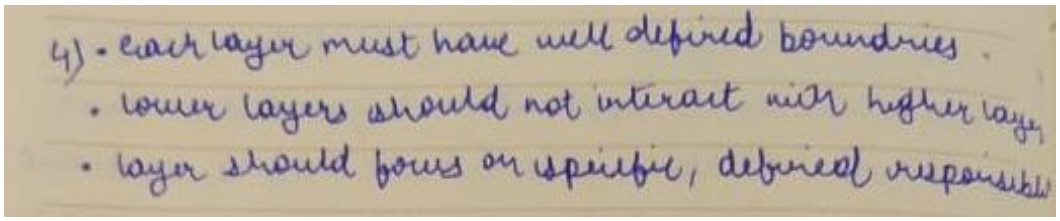
2. What are the allowable structural patterns?


2) • The system is organised into hierarchical layers where each layer serves a distinct function. Each layer interacts only with one directly below it, maintaing strict sepration of concerns.
• Commar structural patterns
  ◦ Client server pattern for user interaction.
  ◦ Pipline pattern for processing requests sequentially across layers.

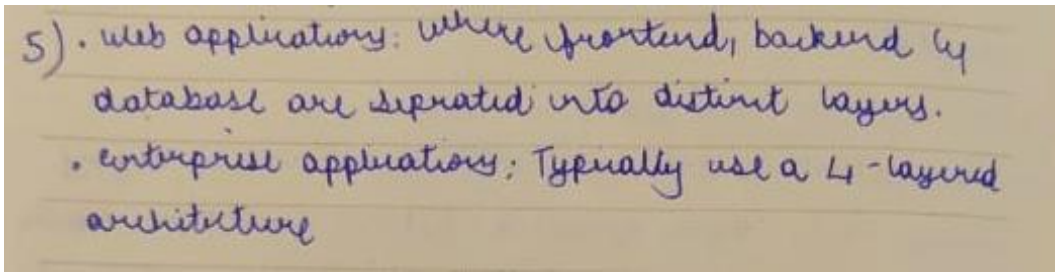
3. What is the underlying computational model?


3) Request - Response Model - The user / client sends request to the topmost layer & each request flows through each layer until it reaches the data layer for processing, after which a response after which a response is sent back in reverse.

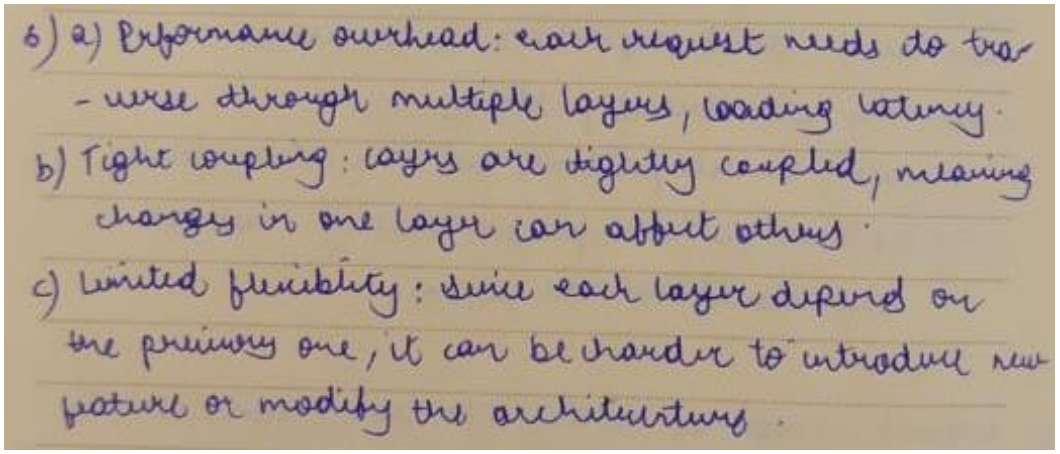
4. What are the essential invariants of the style?


4) • each layer must have well defined boundries.
• lower layers should not interact with higher layer
• layer should focus on specific, defined responsibil

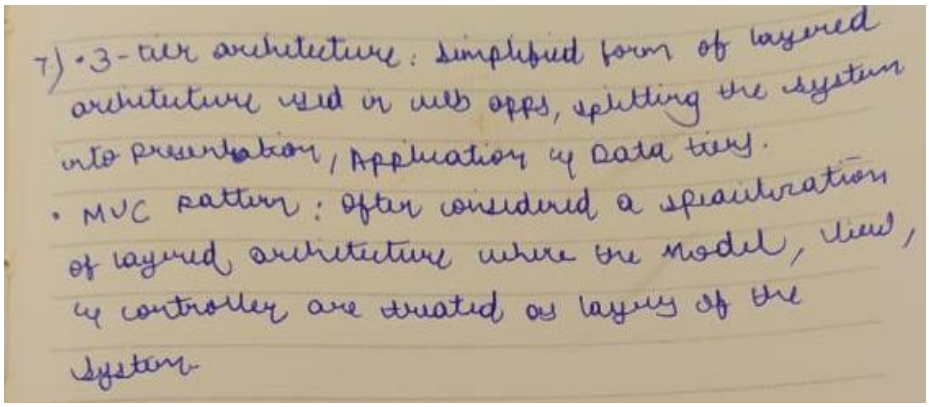
5. What are common examples of its use?


5). web applications: where frontend, backend & database are seprated into distinct layers.
• enterprise applications: Typically use a 4-layered architecture

6. What are the style's specializations?


6) a) Preformance overhead: each request needs do tra- -verse through multiple layers, leading latency.
b) Tight coupling: layers are tightly coupled, meaning changes in one layer can affect others.
c) Limited flexibility: since each layer depend on the previous one, it can be harder to introduce new feature or modify the architecture.

7. What are the styles specifications


7) • 3-tier architecture: simplified form of layered architecture used in web apps, splitting the system into presentation, Application & Data tiers.
• MVC pattern: often considered a specialization of layered architecture where the model, view, & controller are treated as layers of the system.

Date: _____ **Signature of faculty in-charge**