TRAINING REPORT

A Training Report submitted in the partial fulfillment of the requirements for the award of the degree of

# Bachelor of Engineering In

**Information Technology Engineering**



## UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY PANJAB UNIVERSITY, CHANDIGARH

Undertaken at



**Under the guidance of:**

Richa Kaushik [richa.kaushik@incedoinc.com](mailto:richa.kaushik@incedoinc.com) Talent Acquisition Executive Cell: 9078724958

**Submitted by:**

Nikunj Kabra UE198064

IT(4th year)

# CANDIDATE’S DECLARATION

I hereby declare that the project work entitled **“Client Advisor login page”** is an authentic record of my own work carried out at **Incedo** as requirements of six months project semester for the award of degree of

B.E. at University Institute of Engineering and Technology, Punjab University, Chandigarh under the guidance of **Mrs. Richa Kaushik** and **Dr. Preetika Sharma** and **Dr. Deepak Kumar.** During **9 January 2022 – June 2022**.

**Nikunj Kabra UE198064**

*[Digital signatures will be provided by Incedo Inc. together for all interns hired via on campus hiring on request by IT UIET.]*

**Mentor at Incedo Inc.**

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have helped me in completing this report on Incedo Inc. I would like to thank my college and my sincere gratitude to Prof. J.K. Goswamy, Director, UIET and Mr. Rajneesh Singla, Coordinator, IT, UIET for providing me with the opportunity to learn and explore the field of technology solutions.

I would also like to express my heartfelt thanks to the team at Incedo Inc. for their support and cooperation in providing me with the necessary information and insights about their organization. Their prompt responses and valuable inputs were instrumental in helping me complete this report. I would like to express my sincere gratitude to **Mr. Srikumar Pandit** (Senior Manager), **Ms. Aashna Kapoor** (Senior Executive), **Ms. Richa Kaushik** (Talent Acquisition Executive), **Ms. Sumeet Madaan** (Senior Director, Human Resources) at Incedo Inc. and **Dr. Preetika Sharma** for their constant guidance in the ongoing term from **09/01/23 to 06/23.**

Finally, I would like to thank my family and friends for their constant encouragement and support throughout my academic journey.

### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ACKNOWLEDGEMENT** | **3** |
|  | **TABLE OF CONTENTS** | **4** |

### About Incedo Inc. 6

* 1. [Introduction 6](#_bookmark0)
  2. [Horizontals 8](#_bookmark1)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1.2.1 | Data and Analytics | 8 |
|  | 1.2.2 | Digital and Cloud | 8 |
|  | 1.2.3 | Emerging Technologies | 8 |
| 1.3 | Verticals. |  | 9 |
|  | 1.3.1 | Financial Services | 9 |
|  | 1.3.2 | Healthcare and Life Sciences | 9 |
|  | 1.3.3 | Telecom and Hi-Tech | 9 |

1. **Engineering Primer 11**
   1. Introduction 11
   2. [Day 1 Agile 11](#_TOC_250003)
   3. Day 2 The Software Development Lifecycle 13
   4. [Day 3 Software Architecture 14](#_TOC_250002)
   5. Day 4 Software Testing 15
   6. Day 5 Coding Guidelines 16
   7. Day 6 Design Patterns 17
   8. Day 7 Databases 18
2. QA Training 20
   1. Introduction 20
   2. Fundamentals of testing 20
   3. Lifecycle 21
   4. Static Testing 22
   5. Dynamic Testing 23
3. [**Week capstone project** 24](#_TOC_250001)
   1. [Introduction 24](#_TOC_250000)
   2. Progress till now 26
   3. To do next 26

**5. Aligning to business**  **28**

5.1 Introduction 28

5.2 Udemy course 28

* 1. Design pattern for selenium automation 31

**6. Automation and Testing of a website 33**

6.1 Introduction 33

6.2 Work done 33

**7. Conclusion 41**

**8. References 42**

**Chapter 1**

**Introduction**

* 1. **Introduction:**

The company I interned with is Incedo Inc., a leading technology solutions provider headquartered in Santa Clara, California, with additional offices located in India and the United States. The company was founded in 2011 by a team of industry veterans who shared a vision of creating a technology services firm that would help clients transform their business with innovation and digitalization. Incedo provides end-to-end solutions in the areas of data and analytics, digital and cloud, and emerging technologies. The company's services are designed to help clients achieve their business goals by leveraging the latest technologies and best practices in software development, data engineering, machine learning, and digital transformation.

The company's clientele spans across a wide range of industries, including financial services, healthcare, life sciences, telecom, and hi-tech. Incedo has established itself as a trusted partner for its clients, delivering innovative solutions that address complex business challenges while ensuring timely delivery and quality.

Incedo's team comprises over 4,000 highly skilled professionals who bring extensive industry experience and domain expertise to every project. The company places a strong emphasis on building a diverse and inclusive workforce that fosters a culture of collaboration, innovation, and learning. In addition to its core services, Incedo is also committed to social responsibility and community service. The company has established several initiatives to support education, health, and social welfare programs in the communities where it operates.

Overall, Incedo Inc. is a leading technology services provider that is committed to helping clients achieve their business goals with innovative solutions that leverage the latest technologies and best practices. An internship with Incedo would offer valuable learning opportunities in a dynamic and fast- paced work environment, working alongside experienced professionals on cutting-edge projects.

Mission**:** Enable our clients to maximize business impact from technology by:

* + - Harnessing the transformational impact of emerging technologies.
    - Bridging the gap between business and technology.
    - Become an employer of choice by being ‘employee first’ in all processes and practices.

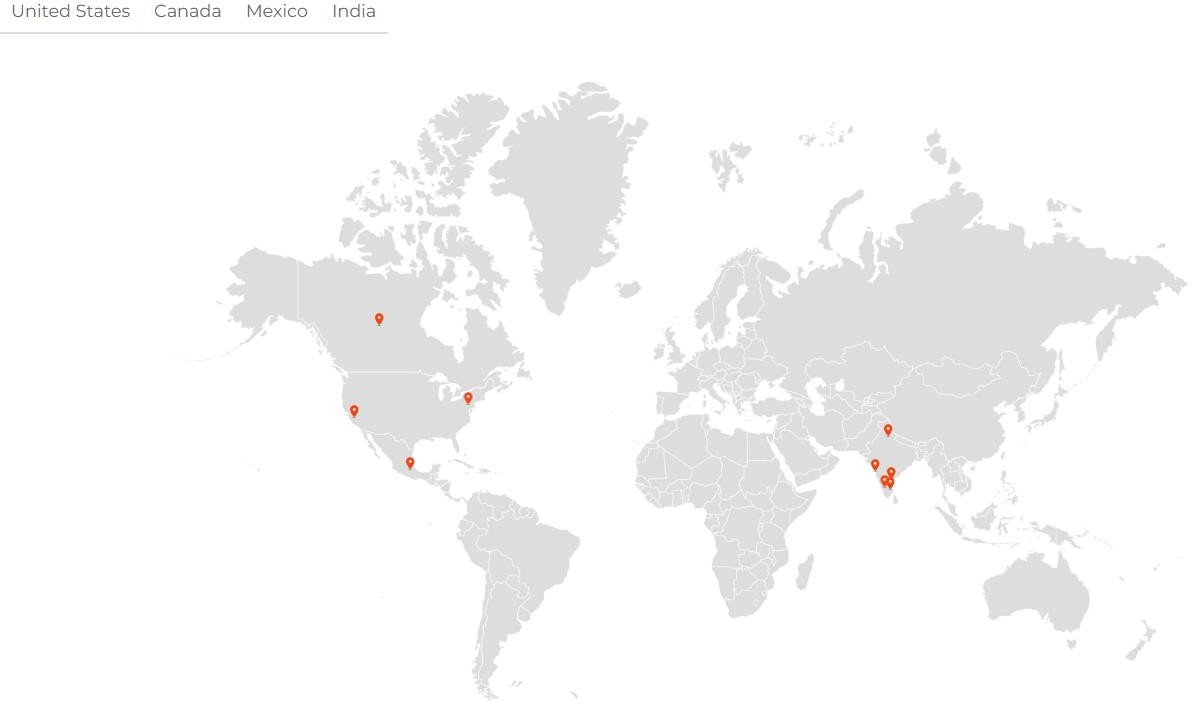
Vision: A World-Class Technology Services Firm that is a trusted, long-term partner for global enterprises, is recognized as an industry leader in chosen emerging technologies, is an employer of choice, and delivers superior growth and financial performance.

The company has prestigious clients all over the world. Some of these clients are US Bank, Citi Bank, Cisco, Belden, Verizon, Tripwire, Symes health, Pfizer, Genentech and much more.

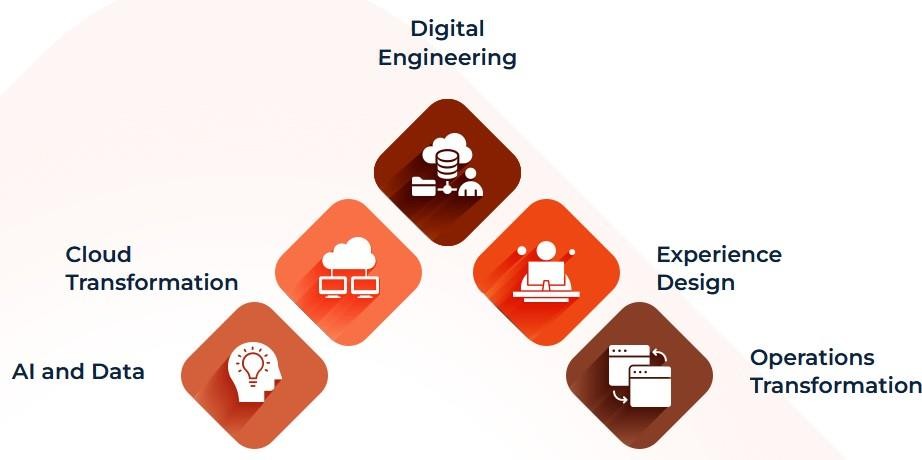
The company operates with 8 core values:

1. Exceed Client Expectations
2. Pursue Excellence
3. Build for the long term
4. Embrace change and Innovation
5. Work as one Global team
6. Be a Caring Meritocracy
7. Drive commercial rigor
8. Always act with integrity

### Office locations across the globe:



* 1. **Horizontals:**

Incedo offers a range of horizontal services that cut across various domains and industries. These services are designed to help clients transform their businesses by leveraging technology. The horizontals offered by Incedo are:

*Figure 1.1: Incedo Horizontals*

* + 1. **Data and Analytics:** Incedo's data and analytics services help clients unlock the power of data to gain insights into their business operations. The company offers services in data engineering, data science, data governance, and data visualization. Incedo's data engineering services help clients transform and optimize their data architecture, while the data science services provide advanced analytics to help clients make better decisions. The company's data governance services help clients manage their data assets effectively, and the data visualization services help clients communicate insights effectively.
    2. **Digital and Cloud:** Incedo's digital and cloud services help clients transform their business operations by adopting digital technologies and cloud computing. The company offers services in digital strategy, user experience design, software engineering, cloud migration, and DevOps. Incedo's digital strategy services help clients define their digital roadmap, while the user experience design

services help clients design intuitive and engaging user experiences. The software engineering services provide end-to-end software development services, while the cloud migration services help clients move their existing applications to the cloud. Finally, the DevOps services help clients accelerate their software development and delivery.

* + 1. **Emerging Technologies:** Incedo's emerging technologies services help clients stay ahead of the curve by leveraging new and emerging technologies. The company offers services in artificial intelligence, machine learning, blockchain, and IoT. Incedo's AI and machine learning services help clients build intelligent systems that can automate processes and provide personalized experiences.
  1. **Verticals:** In addition to the horizontals, Incedo also offers specialized services for specific industries. These verticals are:
     1. **Financial Services:** Incedo's financial services vertical provides solutions for banking, capital markets, insurance, and wealth management. The company offers services in regulatory compliance, risk management, data governance, and customer experience. Incedo's financial services solutions help clients manage their regulatory compliance requirements, mitigate risks, and improve customer experience.
     2. **Healthcare and Life Sciences:** Incedo's healthcare and life sciences vertical provides solutions for healthcare providers, payers, and life sciences companies. The company offers services in healthcare analytics, patient engagement, clinical data management, and pharmacovigilance. Incedo's healthcare and life sciences solutions help clients improve patient outcomes, reduce costs, and improve their operational efficiency.
     3. **Telecom and Hi-Tech:** Incedo's telecom and hi-tech vertical provide solutions for telecommunications and technology companies. The company offers services in network engineering, digital transformation, cloud adoption, and product engineering. Incedo's telecom and hi-tech solutions help clients build robust and scalable networks, adopt digital technologies, and launch innovative products.



*Figure 1.2: Incedo Verticals*

Incedo's focus on innovation and customer satisfaction has allowed it to build a strong reputation in the technology solutions industry. The company's commitment to providing customized solutions and staying at the forefront of emerging technologies has enabled it to compete with larger, more established companies in the industry. Overall, my experience with Incedo has been a valuable learning opportunity, and I have gained insights into the inner workings of a successful technology solutions provider.

# Chapter-2 Engineer Primer

* 1. **Introduction:** We went through 7 days of engineering primer as part of our internship where our instructor Neeraj taught us different high-level concepts in software engineer, product management, coding best practices etc. A summary of the 7 days is as follows:

### Day 1 Agile:

Agile is a methodology that emphasizes flexibility, collaboration, and iterative development to deliver high- quality software. The Agile Manifesto, created in 2001, values individuals and interactions, working software, customer collaboration, and responding to change. Agile methodologies aim to enable teams to respond to feedback and changing requirements quickly, efficiently, and with minimal disruption.

Agile frameworks share some common principles, including:

* + 1. Iterative development: Agile teams work in short sprints, typically 2-4 weeks, during which they deliver working software. Each sprint includes planning, development, testing, and review, and teams prioritize the most critical features to be delivered in each sprint.
    2. Continuous feedback: Agile teams regularly seek feedback from customers and stakeholders to ensure that the software being developed meets their needs. This feedback is used to adjust the product backlog and guide development efforts.
    3. Collaboration and communication: Agile teams prioritize face-to-face communication and collaboration between team members and stakeholders. This helps to ensure that everyone is aligned and working towards the same goals.
    4. Continuous improvement: Agile teams seek to continuously improve their processes and products. Retrospectives are held at the end of each sprint to reflect on what went well, what could be improved, and what actions should be taken to improve.

Agile methodologies include several practices and frameworks, including:

1. Scrum: Scrum is an Agile framework that focuses on the delivery of small, incremental improvements. It includes roles such as the Scrum Master, Product Owner, and Development Team, and practices such as daily stand-ups, sprint planning, sprint reviews, and retrospectives.
2. Kanban: Kanban is an Agile methodology that emphasizes visualizing the work, limiting work in progress, and continuously delivering small batches of work. It uses a visual board to track progress and prioritize work items.
3. Extreme Programming (XP): XP is an Agile methodology that emphasizes software quality, continuous feedback, and rapid development. It includes practices such as pair programming, test- driven development, continuous integration, and frequent releases.

Agile methodologies can provide many benefits to software development teams, including:

* 1. Faster delivery of working software: Agile methodologies prioritize delivering working software quickly and efficiently, enabling teams to respond to changing requirements and feedback.
  2. Improved collaboration and communication: Agile methodologies prioritize face-to-face communication and collaboration between team members and stakeholders, helping to ensure that everyone is aligned and working towards the same goals.
  3. Flexibility and adaptability: Agile methodologies enable teams to adapt to changing requirements and feedback quickly and efficiently, minimizing disruption and waste.
  4. Improved product quality: Agile methodologies prioritize software quality and continuous feedback, leading to higher-quality products that better meet customer needs.

However, Agile methodologies also have some potential drawbacks, including:

1. Complexity
2. Uncertainty
3. Difficulty in scaling

Overall, Agile methodologies can be an effective way to deliver high-quality software quickly and efficiently, but they require careful planning, execution, and ongoing improvement to be successful.

### Day 2 The Software Development Lifecycle (SDLC):

The Software Development Lifecycle (SDLC) is a process that software development teams follow to plan, design, build, test, and deploy software. The SDLC provides a structured approach to software development that ensures quality, efficiency, and consistency throughout the development process.

There are typically five phases in the SDLC:

* + 1. Planning: In this phase, the software development team defines the project scope, goals, and requirements. They also identify the resources, timeline, and budget needed for the project.
    2. Design: In this phase, the software development team creates a detailed design for the software, including its architecture, user interface, and data structures. They also create a plan for testing and deployment.
    3. Development: In this phase, the software development team builds the software according to the design specifications. They write code, create databases, and develop user interfaces.
    4. Testing: In this phase, the software development team tests the software to ensure that it meets the requirements and is free of defects. They conduct functional tests, performance tests, and security tests, among others.
    5. Deployment: In this phase, the software development team deploys the software to the production environment. They also provide training and support to users, and monitor the software for any issues or defects.

The SDLC is typically iterative, meaning that each phase can be revisited and refined throughout the development process. For example, if testing reveals defects in the software, the development team may need to revisit the design phase to make changes and improve the software.

There are several different SDLC models that software development teams can follow, including the Waterfall model, the Agile model, and the DevOps model. Each model has its own unique approach to software development, and teams may choose the model that best suits their needs based on factors such as project scope, team size, and development methodology.

The SDLC provides a structured approach to software development that ensures quality, efficiency, and consistency throughout the development process. By following the SDLC, software development teams can develop software that meets the needs of users, is free of defects, and can be deployed to

production environments with confidence.

### Day 3 Software Architecture:

Software architecture is the high-level structure of a software system that defines its components, their relationships, and the principles and guidelines governing their design and evolution over time. Software architecture serves as the foundation for software development, providing a blueprint for the design, development, and maintenance of a software system.

Software architecture can be described using various architectural styles, such as layered architecture, client- server architecture, micro services architecture, and event-driven architecture, among others. Each architectural style has its own set of principles, patterns, and practices for designing and building software systems.

Software architecture typically involves several key components:

* + 1. Components: The building blocks of a software system, such as modules, libraries, and services. Components are designed to be modular, loosely coupled, and reusable, allowing them to be easily integrated into different software systems.
    2. Relationships: The connections and interactions between components, such as communication protocols and data flows. Relationships are designed to be efficient, secure, and scalable, ensuring that the software system can handle different types and volumes of data and traffic.
    3. Patterns: The common approaches and best practices for designing software systems. Patterns help software architects and developers to create high-quality, maintainable, and scalable software systems.
    4. Quality Attributes: The non-functional requirements that define the quality of the software system, such as performance, reliability, security, and maintainability. Quality attributes are designed to be measurable, testable, and achievable, ensuring that the software system meets the needs of its users.

Software architecture is a critical component of software development, as it provides the foundation for the design, development, and maintenance of software systems. Good software architecture can help to ensure that a software system is scalable, maintainable, secure, and efficient, while also being flexible enough to adapt to changing user needs and technological advances over time. To create effective

software architecture, software architects must have a deep understanding of software design principles, patterns, and practices, as well as the technical and business requirements of the software system.

* 1. **Day 4 Software testing:** Software testing is the process of evaluating software systems or applications to determine whether they meet the specified requirements, are free of defects, and are fit for use. Testing is an essential part of the software development process, as it helps to identify and fix defects before the software is released to users.

There are several different types of software testing, including:

* + 1. Functional testing: This type of testing focuses on the functional requirements of the software, such as input/output validation, user interface testing, and workflow testing. Functional testing ensures that the software performs the tasks it was designed to do.
    2. Performance testing: This type of testing focuses on the performance and scalability of the software, such as load testing, stress testing, and endurance testing. Performance testing ensures that the software can handle different types and volumes of data and traffic.
    3. Security testing: This type of testing focuses on the security and integrity of the software, such as vulnerability testing, penetration testing, and security audits. Security testing ensures that the software is protected against different types of threats, such as hacking, malware, and unauthorized access.
    4. Usability testing: This type of testing focuses on the usability and user experience of the software, such as user interface testing, user acceptance testing, and usability testing. Usability testing ensures that the software is easy to use and meets the needs of its users.
    5. Regression testing: This type of testing focuses on the changes made to the software during development, such as bug fixes, patches, and updates. Regression testing ensures that the changes do not introduce new defects or affect the functionality of the software.

Software testing can be done manually or using automated testing tools and frameworks. Automated testing is typically faster, more efficient, and more accurate than manual testing, but requires significant setup and maintenance.

Software testing is an essential part of the software development process, as it helps to ensure that the software is of high quality, meets user needs, and is free of defects. Testing should be performed

throughout the software development lifecycle, from the early design and development stages to the final deployment and maintenance stages. Good testing practices can help to reduce the risk of defects and ensure that the software meets the needs of its users.

* 1. **Day 5 Coding Guidelines:** Coding guidelines, also known as coding standards or coding conventions, are a set of guidelines and best practices that software developers follow when writing code. Coding guidelines are intended to improve the quality, readability, and maintainability of code, and to ensure that code is consistent and easy to understand for other developers.

Coding guidelines can cover a wide range of topics and aspects of coding, including:

* + 1. Naming conventions: Guidelines for naming variables, functions, classes, and other elements of code to ensure that they are clear and easy to understand.
    2. Indentation and formatting: Guidelines for how code should be indented and formatted to improve readability and consistency.
    3. Comments and documentation: Guidelines for adding comments and documentation to code to explain its purpose, functionality, and usage.
    4. Code structure and organization: Guidelines for how code should be structured and organized to improve readability, maintainability, and scalability.
    5. Error handling and exception handling: Guidelines for how to handle errors and exceptions in code to improve reliability and robustness.
    6. Performance and optimization: Guidelines for how to optimize code for performance, including avoiding unnecessary code and using efficient algorithms and data structures.
    7. Security: Guidelines for how to write secure code to prevent vulnerabilities and protect against potential security threats.

Coding guidelines can be specific to a programming language, platform, or development environment. Many software development organizations have their own coding guidelines, which are based on their specific needs, requirements, and development practices.

Following coding guidelines can have several benefits, including:

1. Improving code quality: Consistent and well-structured code is easier to read, understand, and

maintain, which can improve overall code quality.

1. Enhancing code readability: Following guidelines for naming conventions, indentation, and formatting can improve code readability and make it easier to understand.
2. Facilitating collaboration: Consistent coding conventions and practices can make it easier for different developers to collaborate on a project.
3. Increasing efficiency: Following coding guidelines can help to reduce errors and improve efficiency, as developers spend less time trying to understand poorly structured or inconsistent code.
   1. **Day 6 Design Patterns:** Design patterns are reusable solutions to common software design problems that have been proven effective over time. They provide a template for solving a specific problem in a software system, and help to improve the overall quality, maintainability, and scalability of the system.

Design patterns can be grouped into several categories, including:

* + 1. Creational patterns: These patterns are used to create objects and instances of classes in a flexible and efficient way, such as the Factory Method pattern, the Abstract Factory pattern, and the Singleton pattern.
    2. Structural patterns: These patterns are used to organize classes and objects into larger structures, such as the Adapter pattern, the Bridge pattern, and the Facade pattern.
    3. Behavioral patterns: These patterns are used to manage communication between objects and classes, and to improve the overall functionality and behavior of the system, such as the Observer pattern, the Strategy pattern, and the Command pattern.

Design patterns can provide several benefits for software development, including:

1. Reusability: Design patterns are reusable solutions to common problems, which can save developers time and effort when developing new software systems.
2. Scalability: Design patterns provide a way to organize and structure code in a scalable and maintainable way, which can make it easier to add new features and functionality to a system over time.
3. Maintainability: Design patterns provide a clear and well-defined structure for code, which can make it easier to maintain and debug over time.
4. Standardization: Design patterns provide a standard way of solving common software design problems, which can make it easier for different developers to understand and work with code.

However, it is important to note that design patterns should be used judiciously, as overuse or misuse of patterns can lead to code that is overly complex, difficult to understand, and difficult to maintain.

Overall, design patterns are an important part of software development, as they provide a well-defined and proven way of solving common software design problems.

* 1. **Day 7 Databases:** A database is a structured collection of data that is organized and managed in a way that allows for efficient storage, retrieval, and manipulation of data. Databases are used in a wide range of applications, including business, finance, healthcare, education, and government.

Databases consist of one or more tables, which are made up of rows and columns. Each row in a table represents a single record or instance of data, while each column represents a specific attribute or piece of information about that record. The structure and organization of tables in a database is called its schema.

Databases can be categorized into several types, including:

* + 1. Relational databases: These databases store data in tables with predefined relationships between them, and use a structured query language (SQL) to manipulate and retrieve data.
    2. NoSQL databases: These databases are designed to handle large amounts of unstructured or semi- structured data, and use a variety of data models and query languages.
    3. Object-oriented databases: These databases store data as objects, and use object-oriented programming concepts to manipulate and retrieve data.
    4. Graph databases: These databases store data in a graph-like structure, and use graph theory algorithms to manipulate and retrieve data.

Databases provide several benefits for software development, including:

1. Data management: Databases provide a centralized and structured way to manage and store data, which can improve data accuracy and consistency.
2. Data security: Databases provide a way to secure and control access to sensitive data, which can prevent unauthorized access and protect against security threats.
3. Scalability: Databases can be designed to scale horizontally or vertically, which allows them to handle larger amounts of data and support more users and applications.
4. Data analysis: Databases provide a way to analyses and extract insights from data, which can support decision-making and improve business intelligence.
5. Data integration: Databases can be integrated with other systems and applications, which allows for seamless data exchange and sharing.

# Chapter-3

**Quality Assurance Training**

* 1. **Introduction:** I have been offered role of Quality Assurance engineer, which entitles the task to ensure the quality of the software and systems the company offers to its clients. We went through 5 days of Quality Assurance Training as part of our internship where our instructor Tarun taught us different high-level concepts in software testing. A summary of the 5 days is as follows:
  2. **Day 1 Fundamentals of Testing:** Everything need to be tested even if it is a software so that accuracy must be good and it will last longer. Fault is a state of the software, caused by error. These faults will increase the cost of production if detected after being implemented. So, we need to test all software.

**SEVEN TESTING PRINCIPLES**

Principle 1: Testing Shows Defects Are Present in the Software

As per this principle, testing is a process which shows defects are present is software. Defects are identified by using different software testing execution techniques. At the same time, testing does not prove that after finding defects that there are no defects present in the system. This principle talks about the reduction of several defects in software. There are always chances that the software has undiscovered defects, testing should not be considered as a proof of defect free software.

Principle 2: Exhaustive Testing Is Not Practically Possible

If we talk about this principle, it says it is not possible to test complete software. Test with all combinations of inputs and outputs. i.e., Test with all possible scenarios is not possible. Then you must be thinking how we will test the complete software. See, instead of performing complete or exhaustive testing we go for risk-based testing. Identifying the impact can help us to identify the module which are on high risk.

Principle 3: Start Testing in Early Stage of SDLC

This principle asks to start testing software in the early stage of software development life cycle. This helps in identifying defects and fix them early with low budget and within assigned time period. It

allows to handover software on time with expected quality.

Principle 4: Defects Clustering

Usually, maximum defects in software lie within the limited set of software areas. If you successfully identify this area, it has become quite a simple task for you to bring that sensitive area under the focus of testing. It is considered as one of the most efficient ways to perform testing efficiently.

5th Principle: The Pesticide Paradox

If you are using the same set of test cases repeatedly, then after some time those test cases do not find any new defects. So, it is always recommended to review and revise the test cases on a regular interval in order to find new defects. It is allowed to add new scenario or test cases even after the execution of test set.

6th Principle: Testing is dependent on context

According to this principle; if you are testing web application and mobile application using same testing strategy, then it is wrong. This principle says the testing approach should be different and that is depending on the application. Strategy for testing web application would be different from android mobile app testing.

7th Principle: Absence of errors

This principal points towards the usefulness of the system. In other words, finding the defects and fixing it will not help user unless and until the software is not developed according to the requirement.

* 1. **Day 2 Lifecycle:** Software testing is very vast so we have different software development models.

1. VV&T

Verification: The process of evaluating a system or component to determine whether the products of the given development phase satisfy the conditions imposed at the start of that phase, are we building the product, right? Validation: Determination of the correctness of the products of software development with respect to the user needs and requirements. Are we building the right product?

1. Waterfall Model

It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the for the project which is small and there are no uncertain requirements.

1. V-model

V-model means Verification and Validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development in V-model.

1. RAD model

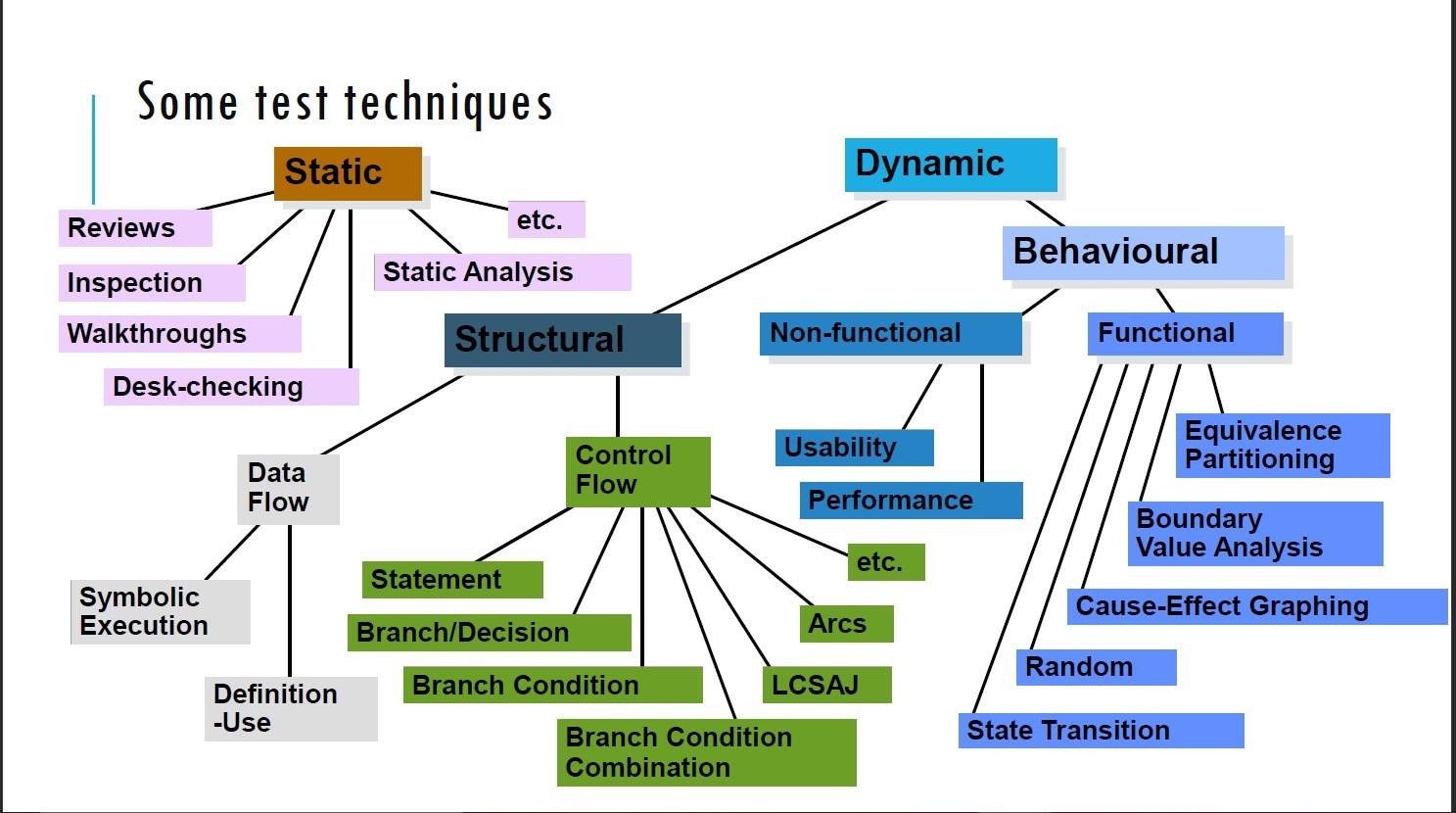
RAD model is Rapid Application Development model. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. Agile Model Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

* 1. **Day 3 Static testing:** It is low level testing in this only review of code or small defects to be seen no automation or coding needed. Static techniques do not execute the code. Just checking the loops in logic, syntax error is part of static testing. Here a tester needs to know the logic and if anything, lag or

need to correct he will inform the developer. Since automation is costly so to save cost and identify little or basic bugs static testing is very essential.

* 1. **Day 4 Dynamic testing:** Doing automation and executing the code is dynamic testing. Exhaustive testing (use of all possible inputs and conditions) is impractical must use a subset of all possible test cases must have high probability of detecting faults. Need thought processes that help us select test cases more intelligently test case design techniques are such thought processes.



**3.5 Day 5 Testing Tools:** For testing we use many software tools for automation testing where we can use a code again and again. It helps us to test the software in all parameters. In this training we learned about Selenium testing tool and Eclipse. Both are currently using in the industry. But there are other tools also like Postman used for API testing, Cypress used for front end testing.

# Chapter-4

# 4 Week Capstone project

### Introduction:

In this we need to make a live project and the project is explained as bellow:

**Project Description**

Creating Advisors and client login, clients can add their investment.

**Project Scope**

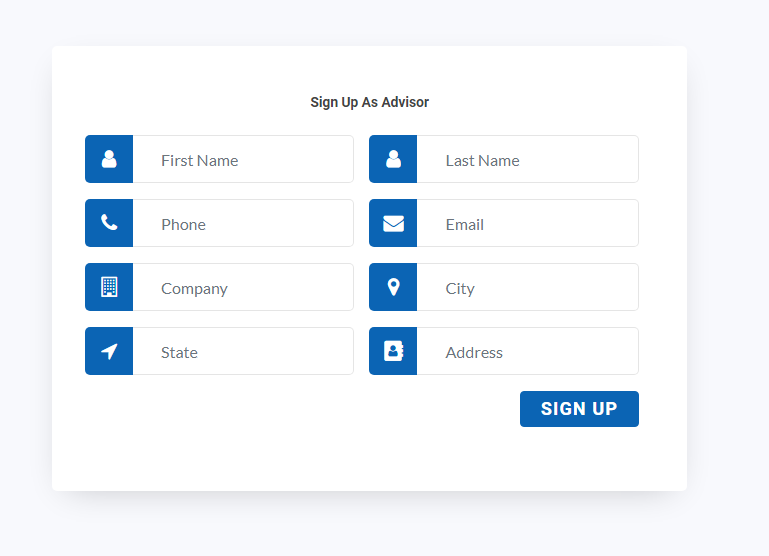
The scope of this project includes the register advisors and clients, and clients can add and update their investment.

**Groups impacted**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **User** | **Responsibilities Impacted** | **System(s) Used** |
| 01 | Advisors | New Advisors and Agents | Demo |
| 02 | Clients | New Clients With investment | Demo |

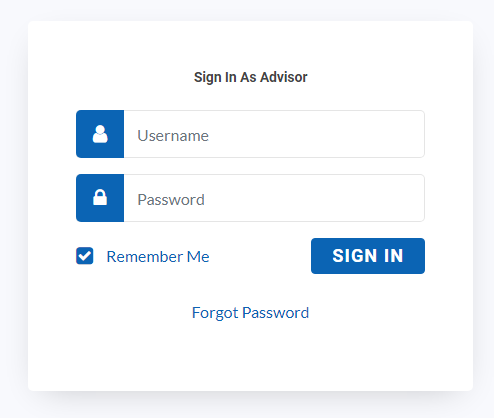
* + 1. **Advisor Register**

Description: - Advisors can register themselves using **[POST]/API/user**

**up**

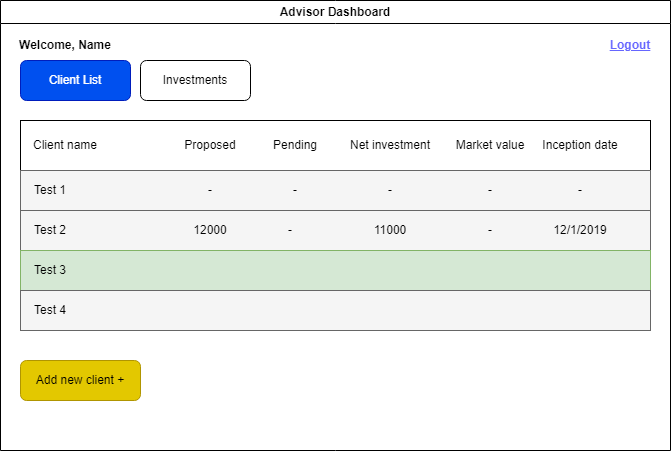
*Figure: Sign up page as advisor*

* + 1. **Advisor Login**

Description: - Advisors can login themselves using **[GET]/API/user/roleID**

*Figure: Sign in page as advisor*

**4.2.1 Advisor dashboard**

Description: - Advisor can edit/delete/add client

### Progress until now:

**Week 1** of the project we worked on gathering the user requirement of the system, basically breaking down the project statement to understand the business needs. As per the business needs figma designs were created to make a roadmap towards the front end of the application.

**Week 2** was dedicated to building the infrastructure and architecture of the web page that how the data will be fetched, analyzed, processed and filtered into the system. High level implementation of data modelling, ER diagram was done to ensure that the roots of the web page that is the data model, entity relations can withstand all sorts of queries and data manipulation and relations.

**Week 3** was dedicated to the database testing and backend development, wherein my personal role as a QA engineer was important to ensure rigorous database testing to ensure that all the data being fetched is correct, data is being fetched, and every parent entity is mapped with its respective children.

**Week 4** was assigned to final evaluation of the work done so far and final quality tests that were to be performed on the application developed so far that is the database- which was already done and the backend for which we use Postman API- a tool to check whether the data being fetched through the API code is same as that of a query run on MySQL server.

Alongside the application development timeline as a QA engineer, we are supposed to maintain a test plan dynamic to the development process which aims as testing the key basic business requirements and all those key areas which could lead to application failure or loss of business. This process of maintaining test plan and running individual test case is called as Manual Testing.

### To do next: 1. Making Frontend dashboards using ReactJS

React is a free and open-source front-end JavaScript library for building user interfaces based on components. In React, you develop your applications by creating reusable components. These components are individual pieces of a final interface, which, when assembled, form the application’s entire user interface.

The ReactJS framework combines the speed and efficiency of JavaScript with a more efficient method of manipulating the DOM to render web pages faster and create highly dynamic and responsive web applications. React relies on a virtual DOM, which is a copy of the actual DOM.

Reacts virtual DOM is immediately reloaded to reflect this new change whenever there is a change in the data state.

### Integration of Frontend with Backend

Integration of Frontend with Backend so that data fetched from database is shown on the frontend screens and requests made from the frontend to update, read, delete, create data is done using backend API’s making a complete working web application.

### Database Normalization

Normalizing databases by making a few tables and linking them using Primary and Foreign keys and thus making a relation between them. This optimizes the operation of working on a database.

### Deployment

After the website is ready, deploy it on a cloud so that anybody can access the website and use it according to their need.

**Chapter 5**

**Aligning To Business**

**5.1 Introduction:** AssetMark, Inc. company’s mission is centered around helping financial advisors make a difference in the lives of their clients. To help them do that, they aim to provide advisors with holistic support. Whether that is through compelling technology that facilitates a better client experience, consulting services that ensure advisors’ businesses are running at their best or offering a comprehensive suite of investment solutions. Asset Mark’s platform empowers advisors to provide the highest level of service possible to their clients.

**5.2 Udemy Course:** We have given a Udemy course to learn more about JAVA and Selenium the course name is “Selenium 4 WebDriver with JAVA (Basics + Advance + Architect)”. In the course we go through various section for our learning such as:

* About selenium.
* Core java in depth for automation.
* Selenium WebDriver basics.
* Selenium 4 handling elements and various locators.
* Xpath and CSS locators in depth.
* Selenium 4 capturing screen shot.

**5.2.1 About selenium: Selenium** is a free (open source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc. to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as **Selenium Testing.**

**A picture containing text, screenshot, font, diagram

Description automatically generated**

**5.2.2 Core java in depth for automation:** The word **Core** describes the basic concept of something, and here, the phrase 'Core Java' defines the basic Java that covers the basic concept of Java programming language. We all are aware that Java is one of the well-known and widely used programming languages, and to begin with it, the beginner has to start the journey with Core Java and then towards the [Advance Java](https://www.javatpoint.com/what-is-advance-java). The [Java programming language](https://www.javatpoint.com/java-tutorial) is a general-purpose programming language that is based on the OOPs concept.

**The following concepts are some of the major basic concepts of core Java through which we go through:**

* Java Fundamentals
* [OOPs Concepts](https://www.javatpoint.com/java-oops-concepts)
* [Overloading](https://www.javatpoint.com/method-overloading-in-java) & [Overriding](https://www.javatpoint.com/method-overriding-in-java)
* [Inheritance](https://www.javatpoint.com/inheritance-in-java) with [Interface](https://www.javatpoint.com/interface-in-java) and [Abstract Class](https://www.javatpoint.com/abstract-class-in-java)
* [Exception Handling](https://www.javatpoint.com/exception-handling-in-java)
* [Packages](https://www.javatpoint.com/package)
* [Collections](https://www.javatpoint.com/collections-in-java)

**5.2.3 Selenium WebDriver basics:** Selenium WebDriver is a web framework that permits you to execute cross-browser tests. This tool is used for automating web-based application testing to verify that it performs expectedly.

Selenium WebDriver allows you to choose a programming language to create test scripts. As discussed earlier, it is an advancement over Selenium RC to overcome a few limitations. Selenium WebDriver is not capable of handling window components, but this drawback can be overcome by using tools like Sikuli, Auto IT, etc.

A picture containing text, font, handwriting, rectangle

Description automatically generated

**5.2.4 Selenium 4 handling elements and various locators:** A locator is a way to identify elements on a page. It is the argument passed to the [Finding element](https://www.selenium.dev/documentation/webdriver/elements/finders/) methods.

**A screenshot of a computer

Description automatically generated with low confidence**

After completing the required section for our Automation testing training, we go through the assignments which was there in the Udemy course.

**5.3 Design Pattern for Selenium Automation:** After completing the required section from the Udemy course for Selenium Automation, We the receive the study material for the design pattern for selenium automation which is Page Object Model (POM).

We went through four hours of learning material to understand Page Object Model Framework which includes various new terms which are important for selenium Automation such as:

**5.3.1 Page Object Model (POM): Page Object Model (POM)** is a design pattern, popularly used in test automation that creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance.

Under this model, for each web page in the application, there should be a corresponding Page Class. This Page class will identify the Web Elements of that web page and contains Page methods which perform operations on those Web Elements.

**5.3.2 TestNG:** [TestNG](https://testng.org/doc/) is an open-source test automation framework for Java. It is developed on the same lines as JUnit and NUnit. A few advanced and useful features provided by TestNG make it a more robust framework than its peers. The NG in TestNG stands for ‘Next Generation’. TestNG provides advanced features such as annotations, data-driven testing, test sequencing, and parallel testing to help you organize and execute your Selenium tests more efficiently and effectively.

Benefits of TestNG:

* Group test cases into logical units, making managing and maintaining your test suite easier.
* Run tests in parallel, significantly reducing the time it takes to execute your test suite.
* TestNG provides a wide range of annotations that you can use to customize your tests, such as **@BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest, @BeforeMethod, and @AfterMethod**.
* It supports data-driven testing, allowing you to run the same test case with multiple test data sets.
* Better reporting and logging features than other testing frameworks make identifying and debugging issues in your tests easier.

**5.3.2 Cucumber:** **Cucumber** is a testing tool that supports Behavior Driven Development (BDD). It offers a way to write tests that anybody can understand, regardless of their technical knowledge. In BDD, users (business analysts, product owners) first write scenarios or acceptance tests that describe the behavior of the system from the customer’s perspective, for review and sign-off by the product owners before developers write their codes. Cucumber framework uses [Ruby programming](https://www.guru99.com/ruby-on-rails-tutorial.html)

## Advantages of Cucumber Software

* It is helpful to involve business stakeholders who can’t easily read code
* Cucumber Testing tool focuses on end-user experience
* Style of writing tests allow for easier reuse of code in the tests
* Quick and easy set up and execution
* Cucumber test tool is an efficient tool for [testing](https://www.guru99.com/software-testing-introduction-importance.html)

**Chapter 6**

**Automation and Testing of a website**

**6.1 Introduction:** We have been instructed to choose a website of our choice and done the automation testing of the same using page object model by creating a maven project in eclipse using selenium with java.

We must automate and test at least 3-4 pages of the website. Which includes:

1. Opening the browser with the website URL.
2. Login with the valid username and password.
3. Search for the product we want
4. Select the product for the searched results
5. Click on the buy now button
6. Continue till the payment page.

**6.2 Work done:** For the given problem statement I have choose Dell website for automation testing.

**Following are the snapshots of Dell automation testing:**

A screenshot of a computer

Description automatically generated with medium confidence

Package Explorer

A screen shot of a computer program

Description automatically generated with low confidence

LoginPage.java

A screen shot of a computer

Description automatically generated with medium confidence

LoginPage.java

A screen shot of a computer program

Description automatically generated with low confidence

HomePage.java

A screenshot of a computer program

Description automatically generated with medium confidence

HomePage.java

A screenshot of a computer program

Description automatically generated with medium confidence

ProductPage.java

A screen shot of a computer program

Description automatically generated with low confidence

ProductPage.java

A screenshot of a computer program

Description automatically generated with medium confidence

CoursePage.java

A screen shot of a computer program

Description automatically generated with low confidence

CoursePage.java

A screenshot of a computer program

Description automatically generated with medium confidence

CartPage.Java

A screenshot of a computer program

Description automatically generated with medium confidence

TestBase.java

A screenshot of a computer

Description automatically generated with medium confidence

TestBase.java

A screenshot of a computer

Description automatically generated

Test

A screen shot of a computer

Description automatically generated with medium confidence

Test

# Conclusion:

I am grateful for the chance to work as Quality Assurance intern for a reputable company like Incedo. I have completed almost two months with this organization, which has been a valuable experience for me. I have developed my soft skills, technical skills and enhanced my understanding abilities. The mentors gave informative sessions and guided me through them to ensure that I understood everything. Getting hands-on experience and working on live projects with experienced staff has undoubtedly improved my skills and assisted me in reaching another achievement. I am excited to put what I have learned to good use to achieve my career goals.

**References**

* <https://www.agilealliance.org/agile101/>
* <https://www.scrum.org/resources/what-is-scrum>
* <https://dev.mysql.com/doc/>
* <https://refactoring.guru/design-patterns>
* <https://www.geeksforgeeks.org/socket-programming-in-c-cpp/>
* <https://www.linuxjournal.com/article/2912>
* <https://www.geeksforgeeks.org/inter-process-communication-ipc/>
* <https://ieeexplore.ieee.org/document/4357692>
* [https://www.techopedia.com/2/29461/software/software-development/software- architecture-the- difference-between-architecture-and-design](https://www.techopedia.com/2/29461/software/software-development/software-architecture-the-difference-between-architecture-and-design)
* <https://www.selenium.dev/>
* <https://cucumber.io/>
* [www.google.com](http://www.google.com)
* <https://testng.org/>