



# **Chhatrapati Shahu Ji Maharaj University**

## **Project Report**

on

## **Urbanic Clone**

Submitted in Partial Fulfillment of the Requirements for the

Degree of

## **Bachelors of Computer Application**

By

**Devanshi Mishra**

**[0302532]**

Under the Supervision of

**Mr. Abhay Singh Bhadauria**



## **PSIT College of Higher Education**

**Kanpur-Agra-Delhi National Highway – 2, Bhauti, Kanpur**

**(2022-2023)**

## **DECLARATION**

I hereby declare that the project entitled “**Urbanic Clone**” submitted for the Bachelor of Computer Application degree is my original work and the project has not formed the basis for the award of any other degree of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature of the student

Devanshi Mishra

0302532

BCA

PSIT College of Higher Education, Kanpur

## **CERTIFICATE**

This is to certify that project entitled “**Urbanic Clone**” submitted for partial fulfillment of the degree of BCA under the Department of Bachelor of Computer Application to through PSIT College of Higher Education, Kanpur, done by **Ms. Devanshi Mishra**, Roll No. **0302532** is an authentic work carried out by me under the guidance of **Mr. Abhay Singh Bhadauria**. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Internal Examiner/Guide

External Examiner

Head of Department

## **ACKNOWLEDGEMENT**

Presentation inspiration and motivation have always played a key role in the success of any venture.

I express my sincere thanks to **Dr. APS Bhadauria, Dean, PSIT College of Higher Education, Kanpur.**

I pay my deep sense of gratitude to **Dr. Pragati Upadhyay (HOD)** of BCA Department, **PSIT College of Higher Education** to encourage me to the highest peak and to provide me the opportunity to prepare the project. I am immensely obliged to **my friends** for their elevating inspiration, encouraging guidance and kind supervision in the completion of my project.

I feel to acknowledge my indebtedness and deep sense of gratitude to my guide **Mr. Abhay Singh Bhadauria** whose valuable guidance and kind supervision given to me throughout the course which shaped the present work as its show.

Last, but not the least, **my parents** are also an important inspiration for me. So, with due regards, I express my gratitude to them.

## **ABSTRACT**

An Urbanic clone is a software application that replicates the functionality of the popular Urbanic e-commerce platform. The Urbanic clone is designed to provide users with a similar shopping experience to the original platform, allowing them to browse, select, and purchase products online.

The Urbanic clone includes features such as user registration and login, product listings, shopping cart management, payment processing, and order tracking. The clone is designed to be user-friendly and intuitive, with a clean and modern interface that is optimized for mobile and desktop devices.

The clone is developed using modern web technologies such as HTML, CSS, JavaScript, and PHP. It utilizes a database to store user and product information and leverages APIs to integrate with payment gateways and shipping providers.

The objective of the Urbanic clone is to provide a platform for small businesses and entrepreneurs to create their own e-commerce store, without the need for extensive technical knowledge or resources. By offering a turnkey solution, the Urbanic clone aims to lower the barrier to entry for e-commerce and enable more people to start their own online business.

Overall, the Urbanic clone represents an innovative solution for businesses looking to enter the e-commerce space, offering a comprehensive set of features and a user-friendly interface at an affordable price point.

## **INDEX**

	Page Number
• Chapter-I:	
• Introduction	1
• Existing System	2
• Problems in Existing System	3
• Proposed System	4
• Objective of the Project	5
• Chapter-II:	6-21
• Modules & Description of the Modules	7
• System Requirements	8
• Technology used in project	9
• Architecture used in project	21
• Chapter-III:	22-41
• Feasibility Study not	23
• Software Development Life Cycle	30
• Data Flow Diagram	36
• Chapter-IV:	42-66
• Data Base Schema Design (Note first explain ER Diagram then Schema)	43
• Screen/Snap-Shots of the project	55
• Codding as use in database connectivity	60
• Conclusion	68
• Future Scope	69
• References	70

# **CHAPTER-1**

---

## **INTRODUCTION**

The Urbanic clone is a web-based e-commerce platform that provides a turnkey solution for businesses and entrepreneurs looking to start their own online store. This clone replicates the functionality and user experience of the popular Urbanic e-commerce platform, which is known for its user-friendly interface and modern design.

With the Urbanic clone, businesses can create their own online store without the need for extensive technical knowledge or resources. The clone includes a range of features such as product listings, shopping cart management, payment processing, and order tracking, making it easy for businesses to manage their e-commerce operations.

The Urbanic clone is built using modern web technologies such as HTML, CSS, JavaScript, and PHP. It utilizes a database to store user and product information and leverages APIs to integrate with payment gateways and shipping providers.

The main objective of the Urbanic clone is to provide a comprehensive e-commerce solution for businesses of all sizes, from small startups to large enterprises. By offering a user-friendly interface and a range of features at an affordable price point, the clone aims to lower the barrier to entry for e-commerce and enable more people to start their own online business.

In summary, the Urbanic clone is a powerful tool for businesses looking to enter the e-commerce space, offering a comprehensive set of features and a user-friendly interface that makes it easy to create and manage an online store.



## **EXISTING SYSTEM**

The existing Urbanic e-commerce platform is a popular web-based platform that allows businesses to create and manage their own online stores. The platform is known for its user-friendly interface and modern design, which make it easy for businesses to manage their e-commerce operations.

The Urbanic platform includes a range of features such as product listings, shopping cart management, payment processing, and order tracking. These features are designed to provide a comprehensive e-commerce solution for businesses of all sizes, from small startups to large enterprises.

The Urbanic platform is built using modern web technologies such as HTML, CSS, JavaScript, and PHP. It utilizes a database to store user and product information and leverages APIs to integrate with payment gateways and shipping providers.

One of the main advantages of the Urbanic platform is its ease of use. The platform is designed to be intuitive and user-friendly, making it easy for businesses to create and manage their online stores without the need for extensive technical knowledge or resources.

However, one of the limitations of the Urbanic platform is its cost. While the platform offers a comprehensive set of features, it can be expensive for small businesses and entrepreneurs to use, particularly those who are just starting out.

In summary, the existing Urbanic e-commerce platform is a popular and powerful tool for businesses looking to create and manage their own online stores. While the platform offers a comprehensive set of features and a user-friendly interface, its cost can be a limiting factor for some businesses.

## **PROBLEMS IN EXISTING SYSTEM**

While the existing Urbanic e-commerce platform is a popular and powerful tool for businesses looking to create and manage their own online stores, there are several problems with the system that need to be addressed. These problems include:

1. **Cost:** One of the main problems with the Urbanic platform is its cost. While it offers a comprehensive set of features, the platform can be expensive for small businesses and entrepreneurs to use, particularly those who are just starting out.
2. **Customization:** Another problem with the Urbanic platform is its limited customization options. While the platform provides a range of templates and themes to choose from, businesses may find it difficult to create a unique look and feel for their online store.
3. **Scalability:** As businesses grow and their e-commerce operations expand, the Urbanic platform may not be able to keep up with their needs. The platform may not offer the scalability and flexibility required to manage large volumes of traffic and transactions.
4. **Integration:** While the Urbanic platform integrates with a range of payment gateways and shipping providers, businesses may find it challenging to integrate with other third-party systems and services.
5. **Security:** The Urbanic platform may not offer the level of security required to protect sensitive user and payment information. Businesses may need to implement additional security measures to ensure the safety and security of their e-commerce operations.

In summary, while the Urbanic e-commerce platform offers a range of features and a user-friendly interface, there are several problems with the system that need to be addressed, including cost, customization, scalability, integration, and security.

## **PROPOSED SYSTEM**

The proposed Urbanic clone aims to address the problems in the existing Urbanic e-commerce platform by providing a more affordable, customizable, scalable, integrated, and secure e-commerce solution for businesses.

The proposed system includes the following features:

- 1. Cost-Effective Pricing:** The Urbanic clone will offer more affordable pricing options for businesses of all sizes, including small startups and entrepreneurs who may have limited budgets.
- 2. Customizable Design:** The clone will offer a range of customization options, including customizable templates and themes, allowing businesses to create a unique look and feel for their online stores.
- 3. Scalable Infrastructure:** The Urbanic clone will utilize scalable infrastructure, allowing businesses to manage large volumes of traffic and transactions as their e-commerce operations grow.
- 4. Integration with Third-Party Systems:** The clone will offer seamless integration with a range of third-party systems and services, including marketing tools, accounting software, and customer relationship management (CRM) systems.
- 5. Enhanced Security Measures:** The Urbanic clone will implement advanced security measures to protect sensitive user and payment information, including SSL encryption, two-factor authentication, and fraud detection.

Overall, the proposed Urbanic clone aims to provide a more comprehensive and affordable e-commerce solution for businesses of all sizes, with a focus on customization, scalability, integration, and security. By addressing the problems in the existing Urbanic e-commerce platform, the clone will enable more businesses to start and grow their online stores.

## **Objective of the Project**

The objective of the Urbanic clone project is to develop an e-commerce platform that provides a comprehensive and affordable solution for businesses looking to create and manage their own online stores. The project aims to address the problems in the existing Urbanic e-commerce platform by providing a more customizable, scalable, integrated, and secure solution for businesses.

Specific objectives of the Urbanic clone project include:

1. Developing a user-friendly interface that is easy to navigate and use for both businesses and customers.
2. Providing affordable pricing options for businesses of all sizes, including small startups and entrepreneurs.
3. Offering a range of customization options to allow businesses to create a unique look and feel for their online stores.
4. Utilizing scalable infrastructure to manage large volumes of traffic and transactions as businesses grow.
5. Integrating with a range of third-party systems and services to streamline business operations and increase efficiency.
6. Implementing advanced security measures to protect sensitive user and payment information.
7. Providing comprehensive customer support and technical assistance to ensure that businesses can make the most of the platform's features and capabilities.

Overall, the objective of the Urbanic clone project is to provide a powerful and affordable e-commerce platform that enables businesses of all sizes to create and manage their own online stores with ease and efficiency.

# **CHAPTER-2**

## **MODULES**

The Urbanic clone can be divided into several modules, each of which serves a specific function in the overall e-commerce platform. The modules of the Urbanic clone include:

**1. User Module:** The User Module is the primary module of the Urbanic clone and is responsible for managing user accounts, including account creation, login, and account management. This module includes features such as user profiles, account settings, and order history.

**2. Product Module:** The Product Module is responsible for managing the products available for purchase on the e-commerce platform. This module includes features such as product listings, product descriptions, images, and pricing information.

**3. Order Module:** The Order Module is responsible for managing the order processing and fulfillment process. This module includes features such as shopping cart, checkout process, order confirmation, and order tracking.

**4. Payment Module:** The Payment Module is responsible for managing the payment processing system for the e-commerce platform. This module includes features such as integration with payment gateways, processing of credit card payments, and handling of refunds.

**5. Shipping Module:** The Shipping Module is responsible for managing the shipping and delivery of products to customers. This module includes features such as integration with shipping providers, tracking of packages, and management of shipping rates.

**6. Customer Support Module:** The Customer Support Module is responsible for providing customer support and technical assistance to users of the e-commerce platform. This module includes features such as live chat, email support, and knowledge base articles.

**7. Marketing Module:** The Marketing Module is responsible for managing marketing and promotional activities for the e-commerce platform. This module includes features such as email marketing, social media marketing, and advertising.

**8. Analytics Module:** The Analytics Module is responsible for providing data analytics and insights to businesses using the e-commerce platform. This module includes features such as sales reports, traffic analytics, and customer behavior tracking.

Overall, the modules of the Urbanic clone work together to provide a comprehensive and powerful e-commerce platform for businesses looking to create and manage their own online stores.

## **SYSTEM REQUIREMENTS**

### **SOFTWARE REQUIREMENTS**

- **Development Tool:** -
  - VS code 1.59 and above.
  - X Code 3.2 and above.
- **Operating System:** - Win down 10,11 / Linux
- **Database:** -DB.SQL LITE3
- **Framework:** - Express JS

### **HARDWARE RESOURCE REQUIREMENTS**

- **RAM:** - 1GB or above
- **Memory:** - 500 MB
- **Processor:** - Intel Pentium or above

## **TECHNOLOGY**

**Frontend:** HTML,CSS,JAVASCRIPT,BOOTSTRAP, REACT JS

**Backend:** MONGO DB

**Database:** Node JS

**Framework:** Express JS



## **HTML**

HTML, the fifth major revision of the Hypertext Markup Language (HTML), is a powerful and versatile markup language used for structuring and presenting content on the web. It introduces new features and elements that enhance the capabilities of web pages and offers improved compatibility with modern web browsers and devices. In this response, I will provide you with a comprehensive overview of HTML5, covering various aspects and concepts related to this widely adopted standard.

HTML5 introduces several key features and enhancements, including:

**Doctype Declaration:** HTML introduces a simplified doctype declaration, `<!DOCTYPE html>`, which is backward-compatible with older versions of HTML. This declaration triggers standards mode in web browsers, ensuring consistent rendering of web pages.

**Semantic Elements:** HTML introduces a set of semantic elements that provide meaning and structure to the content. These elements include `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, and `<footer>`. They allow developers to define the purpose and relationships between different parts of a web page, making it more accessible to both humans and search engines.

**Multimedia Support:** HTML provides native support for multimedia content without the need for plugins. The `<audio>` and `<video>` elements allow developers to embed audio and video files directly into web pages, with customizable controls, captions, and responsive playback.

**Canvas Element:** The `<canvas>` element in HTML provides a 2D drawing surface that allows dynamic rendering of graphics, animations, and interactive visualizations using JavaScript. It enables developers to create games, data visualizations, and other rich multimedia experiences directly within the web browser.

**Forms and Input Types:** HTML introduces new input types and attributes for forms, making it easier to capture and validate user input. Input types like email, url, number, date, and range provide enhanced user experiences and facilitate data validation without requiring JavaScript.

**Local Storage:** HTML introduces the local Storage API, which allows web applications to store data locally on the user's device. This feature enables offline capabilities and the ability to cache resources, enhancing the performance and user experience of web applications.

**Geolocation:** HTML provides the Geolocation API, allowing web applications to access the user's location information if the user grants permission. This feature enables location-based services, such as mapping, local search, and personalized content based on the user's whereabouts.

**Web Workers:** HTML introduces web workers, which are background scripts that run concurrently with the main web page. Web workers enable the execution of complex tasks without blocking the user interface, improving the responsiveness and performance of web applications.

**Drag and Drop:** HTML includes native support for drag and drop interactions, making it easier to implement intuitive user interfaces. Developers can enable drag and drop functionality for elements on the web page, allowing users to move or manipulate content effortlessly.

**Responsive Web Design:** HTML incorporates features like media queries and viewport meta tags, enabling developers to create responsive web pages that adapt to different screen sizes and devices. Responsive design ensures optimal viewing experiences on desktops, tablets, and mobile devices.

In conclusion, HTML is a significant advancement in web technology, offering new features and capabilities that enhance the functionality, interactivity, and accessibility of web pages. It has revolutionized the way web content is created and consumed, empowering developers to build

## CSS

CSS, short for Cascading Style Sheets, is a fundamental technology used in web development to style and format the appearance of HTML documents. It provides a set of rules, properties, and selectors that allow developers to define how elements on a web page should be presented to the user. In this response, I will provide you with a comprehensive overview of CSS, covering its key concepts, features, and its role in modern web design.

CSS separates the presentation layer from the content layer of a web page, enabling developers to control the visual aspects of elements without directly modifying the HTML markup. By applying CSS styles, you can change the font, color, size, spacing, layout, and other visual properties of elements, creating visually appealing and consistent designs across multiple web pages.

Here are some essential concepts and features of CSS:

**Selectors:** CSS uses selectors to target specific elements in an HTML document. Selectors can be based on element names, classes, IDs, attributes, or relationships between elements. For example, the selector "h1" targets all the heading elements, while ".class" targets elements with a specific class name.

**Properties:** CSS properties define the visual attributes of elements. Examples include "color" for setting the text color, "font-size" for specifying the font size, "background-color" for setting the background color, and "margin" for controlling the spacing around elements.

**Values:** CSS properties are assigned values that determine the appearance of elements. Values can be specific colors, numerical measurements (e.g., "12px", "1rem"), keywords or a combination of these.

**Styling Methods:** CSS allows styles to be applied in different ways. Styles can be defined inline directly within the HTML elements using the "style" attribute. Alternatively, styles can be placed in a separate CSS file and linked to the HTML document using the "link" element or embedded within the HTML document using the "style" element.

**Cascading and Specificity:** CSS follows a cascading nature, meaning that multiple styles can be applied to an element, and the final appearance is determined by a set of rules. Specificity determines which style rules take precedence when conflicts occur. CSS uses a calculation based on selectors' specificity to determine the most applicable style for an element.

**Box Model:** The box model is a fundamental concept in CSS that defines how elements are rendered in terms of their dimensions and spacing. Each element is represented as a rectangular box with properties like content, padding, border, and margin. Understanding the box model is crucial for controlling the layout and spacing of elements on a web page.

**Responsive Design:** CSS plays a vital role in creating responsive web designs that adapt to different screen sizes and devices. Media queries allow developers to apply different styles based on the device's characteristics, enabling the creation of mobile-friendly and flexible layouts.

**CSS Frameworks and Preprocessors:** CSS frameworks, such as Bootstrap and Foundation, provide pre-designed CSS styles and components that developers can use to build websites quickly. CSS preprocessors like Sass and Less extend CSS with additional features like variables and nested rules, making CSS code more modular and maintainable.

**Transitions and Animations:** CSS supports transitions and animations to add interactive and dynamic effects to web pages. Transition properties enable smooth transitions between different states, while animation properties allow developers to create more complex and customized animations using keyframes.

**Flexbox and Grid Layout:** CSS introduced flexible box layout (Flexbox) and grid layout systems to provide powerful tools for creating responsive and flexible page layouts. Flexbox allows for easy alignment and positioning of elements within a container, while grid layout enables more complex two-dimensional layouts

## **JAVASCRIPT**

JavaScript is a versatile and powerful programming language that is widely used in web development to add interactivity, dynamic behavior, and functionality to websites and web applications. It is a client-side scripting language, meaning it runs directly in the user's web browser. In this response, I will provide you with a comprehensive overview of JavaScript, covering its key concepts, features, and its role in modern web development.

**Syntax and Structure:** JavaScript uses a syntax similar to other programming languages like C, C++, and Java. It supports variables, data types (such as numbers, strings, arrays, and objects), operators, control structures (like loops and conditional statements), functions, and error handling.

**DOM Manipulation:** JavaScript interacts with the Document Object Model (DOM), which represents the structure of an HTML document. It allows developers to manipulate elements on a web page dynamically, changing content, styles, and attributes. With JavaScript, you can create, modify, or remove elements, handle events, and respond to user interactions.

**Event Handling:** JavaScript enables the handling of various events triggered by user actions or other sources. These events include button clicks, form submissions, mouse movements, keyboard inputs, and more. By attaching event listeners to elements, developers can execute specific code when these events occur, providing interactivity and responsiveness to web pages.

**Asynchronous Programming:** JavaScript supports asynchronous programming using techniques such as callbacks, promises, and async/await. Asynchronous operations allow for non-blocking execution, enabling tasks like making API calls, loading resources, and handling user input without freezing the browser's user interface.

**Libraries and Frameworks:** JavaScript has a vast ecosystem of libraries and frameworks that simplify and accelerate web development. Popular JavaScript libraries include jQuery (for DOM manipulation and AJAX requests), React (for building user interfaces), Vue.js (for building reactive web applications), and Angular (a comprehensive framework for building web applications).

**AJAX and APIs:** JavaScript facilitates asynchronous communication with servers using AJAX (Asynchronous JavaScript and XML) requests. With AJAX, web pages can exchange data with a server in the background, allowing for dynamic content updates without reloading the entire page. Additionally, JavaScript enables integration with various APIs, enabling access to external services and data sources.

**Data Manipulation:** JavaScript provides powerful built-in methods and functions for manipulating and working with data. It supports array manipulation, string manipulation, mathematical operations, date and time handling, regular expressions, and more. JavaScript also supports JSON (JavaScript Object Notation), a widely used data format for exchanging structured information.

**Error Handling and Debugging:** JavaScript includes error handling mechanisms to catch and handle exceptions that may occur during script execution. Developers can use try-catch blocks to handle errors gracefully and provide appropriate feedback to users. JavaScript also offers debugging tools and techniques, such as console logging and browser developer tools, to help identify and fix issues in code.

## **React JS**

React.js is an open-source JavaScript library for building user interfaces. It was developed by Facebook and is widely used for creating interactive and dynamic web applications. React.js follows a component-based architecture, allowing developers to break down the user interface into reusable and self-contained components.

Here are some key features and concepts associated with React.js:

- Component-Based
- Virtual DOM
- JSX
- Unidirectional Data Flow
- State Management
- Lifecycle Methods
- React Router
- Redux

React.js has gained popularity due to its declarative syntax, component reusability, and efficient rendering capabilities. It is widely adopted by developers and has a large ecosystem of libraries and tools that complement its functionality.

## **Mongo DB**

MongoDB is a popular open-source NoSQL database management system that is designed to handle large volumes of data. It belongs to the document-oriented database category, which means it stores data in flexible, JSON-like documents instead of traditional table-based relational structures.

Here are some key features and concepts associated with MongoDB:

- 1. Document-Oriented:** MongoDB stores data in flexible and self-contained documents, which are represented using the BSON (Binary JSON) format. A document is a set of key-value pairs, similar to a JavaScript object, and can contain nested structures and arrays. This document-oriented approach allows for easy and dynamic schema evolution.
- 2. Scalability:** MongoDB is designed to be highly scalable, both horizontally and vertically. It supports automatic sharding, which means it can distribute data across multiple servers or clusters to handle increasing data loads. This allows for efficient scaling of both storage and processing power as the application's needs grow.
- 3. Flexible Schema:** MongoDB offers a flexible schema design, allowing documents within a collection to have different structures and fields. This flexibility enables agile development and accommodates evolving data requirements without the need for schema migrations.
- 4. Query Language:** MongoDB provides a powerful and expressive query language that supports a wide range of operations for retrieving, filtering, and manipulating data. It supports a rich set of query operators and aggregation pipelines, allowing for complex and efficient data retrieval.
- 5. Indexing:** MongoDB supports indexing on fields within documents to improve query performance. Indexes can be created on single fields, compound fields, and even arrays, allowing for fast access to specific data subsets. Indexes can significantly speed up data retrieval operations.
- 6. Replication and High Availability:** MongoDB supports data replication, which involves creating copies of data across multiple servers or clusters. This provides fault tolerance and high availability, ensuring that data remains accessible even in the event of server failures.



**7. ACID Transactions:** Starting from MongoDB version 4.0, it introduced support for multidocument ACID (Atomicity, Consistency, Isolation, Durability) transactions. This allows developers to perform multiple operations on multiple documents as part of a single transaction, ensuring data integrity and consistency.

**8. MongoDB Atlas:** MongoDB Atlas is a fully managed cloud database service provided by MongoDB. It offers automatic scaling, built-in security features, automated backups, and monitoring capabilities. MongoDB Atlas simplifies the deployment and management of MongoDB databases in the cloud.

MongoDB is widely used in various applications, including web applications, mobile apps, IoT (Internet of Things) systems, and analytics platforms. Its flexibility, scalability, and ease of use make it a popular choice for developers working with large and dynamic datasets.

## **Node JS**

Node.js is an open-source JavaScript runtime environment that allows developers to run JavaScript code on the server-side. It provides a powerful platform for building scalable and highperformance applications. Here are some key features and concepts associated with Node.js:

- 1. JavaScript on the Server:** Node.js enables developers to use JavaScript not only on the clientside but also on the server-side. This allows for a consistent language and codebase across different layers of an application, simplifying development and promoting code reuse.
- 2. Asynchronous and Non-blocking I/O:** One of the core strengths of Node.js is its event-driven, non-blocking I/O model. It uses an event loop to handle concurrent requests efficiently, allowing multiple I/O operations to run in parallel without blocking the execution of other code. This makes Node.js well-suited for handling large numbers of concurrent connections.
- 3. NPM (Node Package Manager):** NPM is the default package manager for Node.js and provides a vast ecosystem of open-source libraries and modules. It allows developers to easily manage and integrate third-party packages into their applications, saving time and effort in building functionalities from scratch.
- 4. Server-side Web Development:** Node.js is commonly used for server-side web development, where it powers the backend of web applications. It provides a range of features and tools for building APIs, handling HTTP requests and responses, and interacting with databases. Popular web frameworks like Express.js and Koa.js are built on top of Node.js.
- 5. Microservices Architecture:** Node.js is well-suited for building microservices-based architectures, where applications are broken down into smaller, independent services. Its lightweight and modular nature, along with its ability to handle a large number of concurrent requests, makes it an ideal choice for developing microservices.
- 6. Real-time Applications:** Node.js is often used for building real-time applications that require instant data updates and bidirectional communication between clients and servers. With the help of libraries like Socket.IO, developers can easily create real-time chat applications, collaborative tools, and live streaming platforms.
- 7. Command-line Tools:** Node.js provides a robust set of APIs for building command-line tools and utilities. This allows developers to create custom scripts, automation tools, and CLI (Command Line Interface) applications using JavaScript.

**8. Community and Support:** Node.js has a vibrant and active community of developers worldwide. This community contributes to the growth of the Node.js ecosystem by creating and maintaining a wide range of modules, frameworks, and resources. Developers can find extensive documentation, tutorials, and support from the Node.js community.

Node.js has gained significant popularity for its performance, scalability, and versatility. It is widely used by developers to build a variety of applications, including web servers, APIs, realtime applications, microservices, and command-line tools. Its ability to leverage JavaScript on the server-side and its efficient event-driven architecture make it a compelling choice for modern application development.

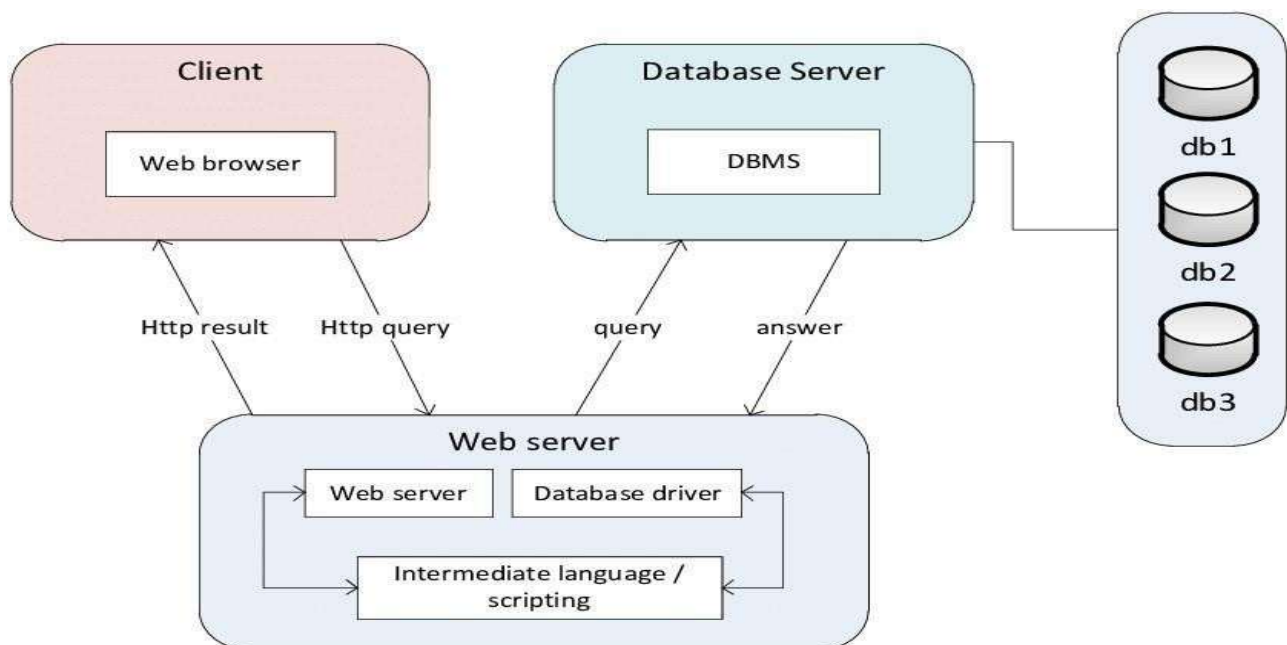
## APPLICATION ARCHITECTURE

### Three-tier architecture

Scholarship Nest uses three-tier architecture. Three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

The user interface is implemented on any platform such as desktop PC, smart phone or tablet as a native application, web app, mobile app, voice interface, etc. It uses a standard graphical user interface with different modules running on the application server.

The relational database management system on the database server contains the computer data storage logic. The middle tiers are usually multitier.



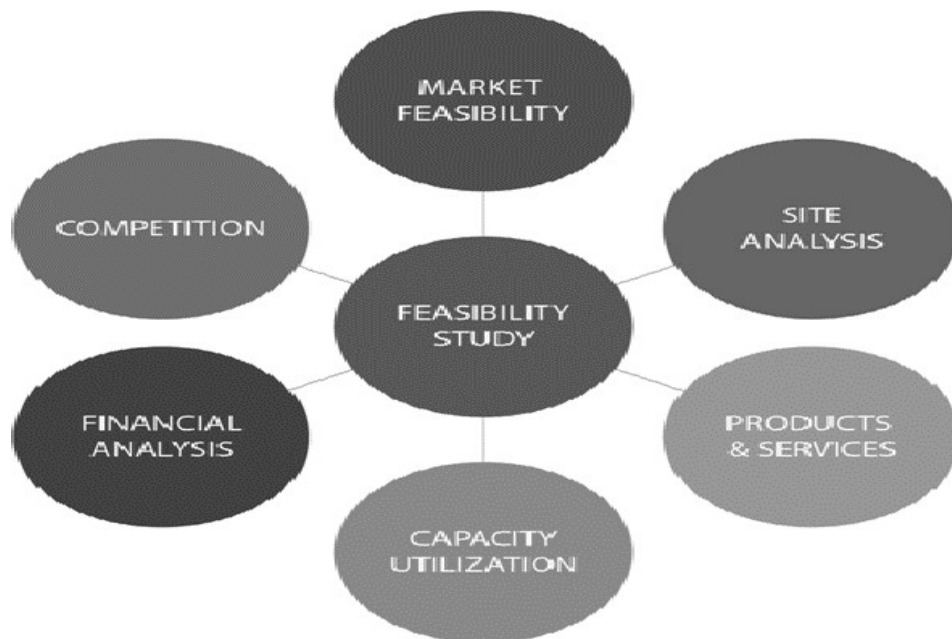
# **CHAPTER-3**

## **FEASIBILITY STUDY**

Feasibility study is a test of a system proposal according to its workability, ability to meet user needs and effective use of resources. The objective of feasibility is not to solve the problem but to acquire a sense of its scope. The main aim of the feasibility study is to test the technical, social and economic feasibility of the system.

Every project is feasible if given unlimited resources and infinite time. Unfortunately, the development of a computer system is more likely to be plugged by a scarcity of resources and difficult delivery dates. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time.

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully.



### **Advantages:**

- The system avoids redundancy by the use of several type of validation that is the system is enhanced.
- Quick access and processing are the main advantage that forces as to implement the proposed system.
- The main alteration between the existing system and the new automated system lies in the specialty which reduces the time consumption in an appropriate manner.
- The system will reduce the amount of paper work require.

Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

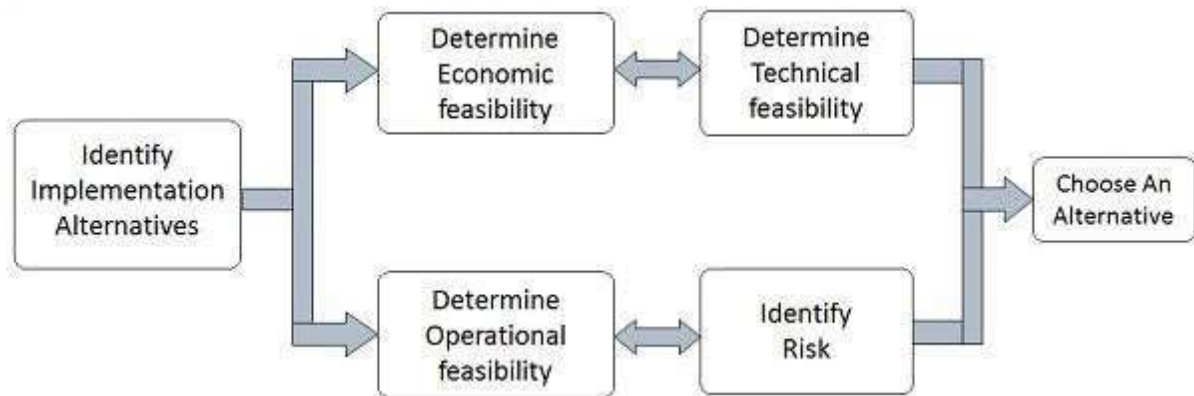
- It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
- It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
- The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
- The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

### **Steps Involved in Feasibility Analysis:**

The following steps are to be followed while performing feasibility analysis –

- Form a project team and appoint a project leader.
- Develop system flowcharts.
- Identify the deficiencies of current system and set goals.
- Enumerate the alternative solution or potential candidate system to meet goals.

- Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
- Weight the performance and cost effectiveness of each candidate system.
- Rank the other alternatives and select the best candidate system.
- Prepare a system proposal of final project directive to management for approval.



The feasibility study can be classified into the following categories:

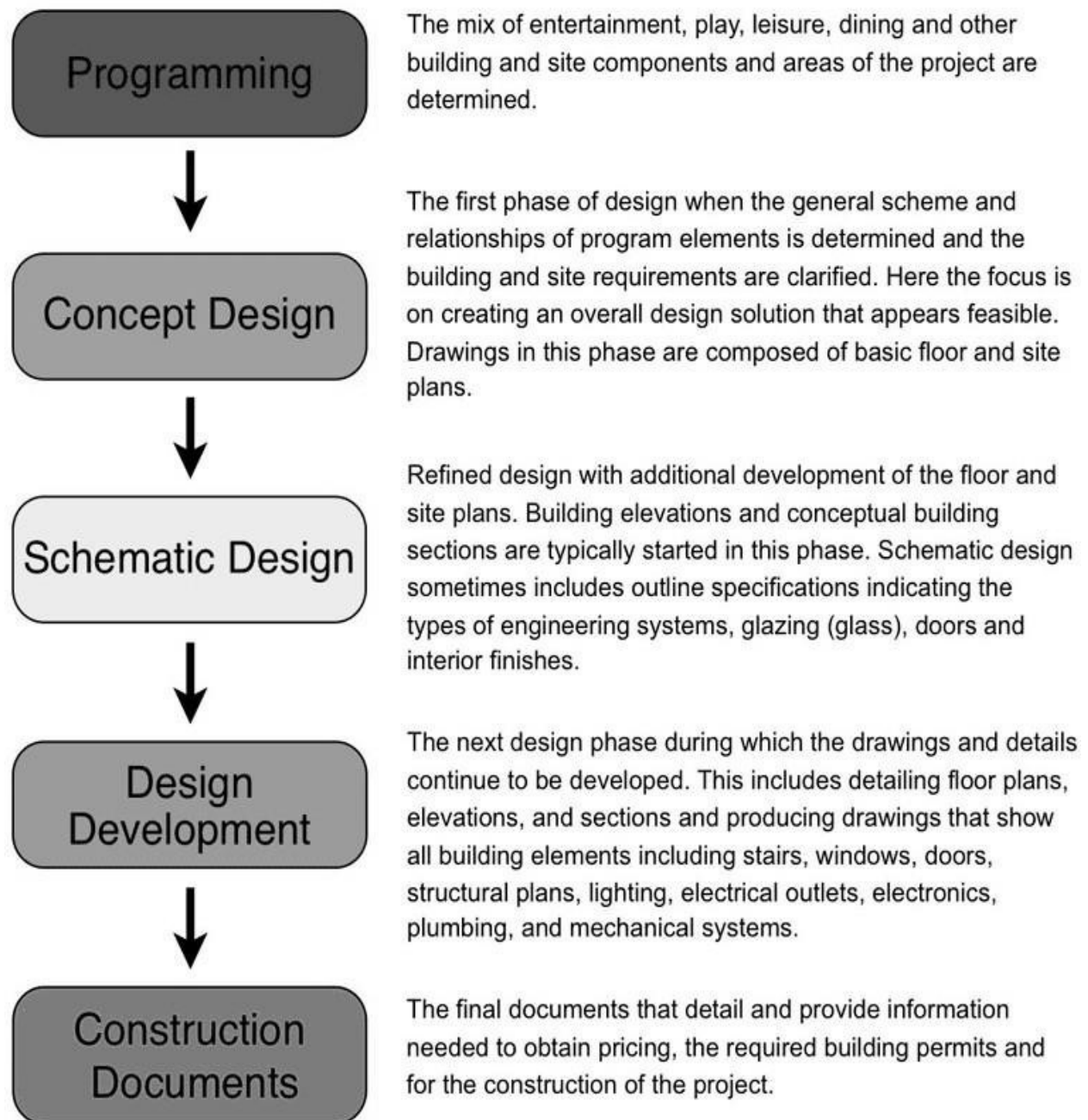
- Economical Feasibility.
- Technical Feasibility.
- Operational Feasibility.

### **Economical Feasibility:**

Since the existing system is manual on the feasibility for wrong data entry is higher and consumes a lot of time and can occur errors. But the proposed system aims at processing of information's efficiently, thus saving the time. The new system need only a system therefore the cost is negligible. Proposed system uses validation check so there will no errors. Even though an initial investment has to be made on the software and the hardware aspects, the proposed system aims at processing of information's efficiently. Hence the benefits acquired out of the system are sufficient enough for the project to be undertaken.



- It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.
- The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
- It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.



## **Tools used to gather information:**

RE techniques, also known as **information gathering** methods/**tools**, are methods used by analysts to determine the needs of customers and users. Techniques that provide safety, utility, usability, learn ability etc. for stakeholders result in their cooperation, commitment and sincerity.

Assessment is about gathering the necessary information about who you are as a literacy agency, your current strengths, weakness, opportunities and threats. Remember, at this point, you are simply gathering information on the current internal and external environment facing your literacy agency; you are not making any judgments or drawing any conclusions about what the information means for the future.

There are many different methods of information gathering that people have used to good advantage and here are a few:

Questionnaires, surveys and checklists: Used when you want to collect a lot of information from people in a non-threatening way.

### 1. Personal interviews

- Used when you want to fully understand a person's opinions or point of view or to get additional information to a questionnaire.

### 2. Documentation review

- Used when you want to gather information on current practices without interrupting the program by examining program monitoring reports, program statistics, learner progress reports, annual reports, performance appraisals, board evaluations, written policies and procedures, memos, minutes, financial records, etc.

### 3. Observation

- Used to watch the program in operation to gather information about what actually happens day-to-day.

### 4. Focus group

- Used to explore a topic in depth with key stakeholders to learn what the common understanding is on various issues.

### 5. Case Studies

- Used to depict experiences, processes or practices with a view to developing understanding through examination and cross comparisons.

**Interview:**

Gather requirements from public by interviews.

**Survey:**

Gather requirements from public in local areas.

**Government:**

Government also want security on travelling people.

**Public:**

Gather requirements from public in local areas by camps.

## **SOFTWARE DEVELOPMENT LIFE CYCLE**

The Software Development Life Cycle (SDLC) is a process used by software development teams to plan, design, build, test, and deploy software. It is a framework that provides a structured approach to software development, with a focus on producing high-quality software that meets the needs of users and stakeholders. There are several stages in the SDLC, each with its own goals and activities. The typical stages of the SDLC are:

1. **Planning:** In this stage, the goals, requirements, scope, and objectives of the software are defined. The team identifies the needs of the stakeholders, and the project plan is created.
2. **Analysis:** The team analyzes the requirements of the software, including any constraints or limitations, to determine the feasibility of the project.
3. **Design:** In this stage, the architecture of the software is designed, including the database schema, the user interface, and the system components.
4. **Development:** The actual coding of the software takes place in this stage. The development team writes the code according to the design, using best practices and following coding standards.
5. **Testing:** In this stage, the software is tested for functionality, performance, security, and other quality metrics. Testing can be automated or manual, and the software is debugged and fixed as necessary.
6. **Deployment:** Once the software is tested and approved, it is deployed to the production environment. This can include installation, configuration, and migration of data.
7. **Maintenance:** After deployment, the software is monitored and maintained to ensure it continues to meet the needs of users and stakeholders. This can include bug fixes, updates, and enhancements.

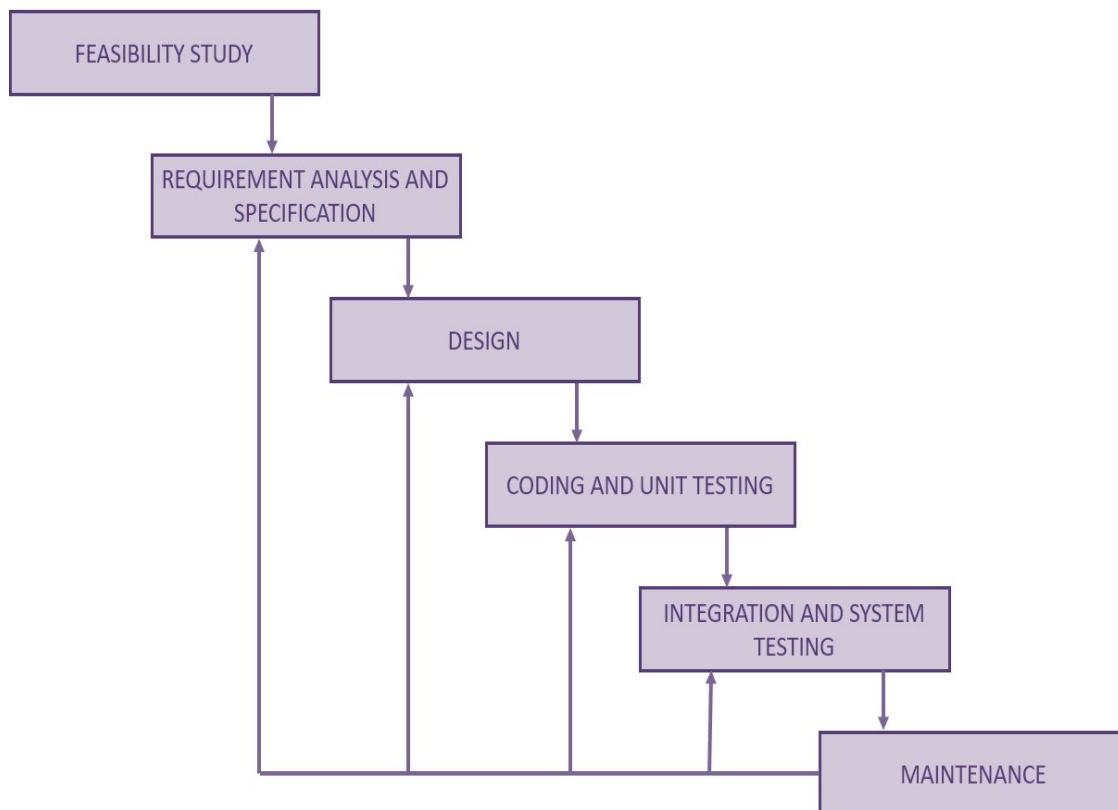
The SDLC is a structured approach to software development that ensures that the software is built efficiently, with high quality and meets the requirements of stakeholders. It is an iterative process, with each stage informing the next, allowing teams to improve the software and the process over time.

### **ITERATIVE WATERFALL MODEL**

In a practical software development project, this is hard to use. So, Iterative waterfall model can be thought of as incorporating the necessary changes to the classical waterfall model to make it usable in practical software development projects. It is almost same as the classical waterfall model except some changes are made to increase the efficiency of the software development.

The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model.

Feedback paths introduced by the iterative waterfall model are shown in the figure below.



When errors are detected at some later phase, these feedback paths allow correcting errors committed by programmers during some phase. The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases. But there is no feedback path to the stage – feasibility study, because once a project has been taken, does not give up the project easily. It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct the errors.

**Phase Containment of Errors:** The principle of detecting errors as close to their points of commitment as possible is known as Phase containment of errors.

### **Advantages of Iterative Waterfall Model**

- **Feedback Path:** In the classical waterfall model, there are no feedback paths, so there is no mechanism for error correction. But in iterative waterfall model feedback path from one phase to its preceding phase allows correcting the errors that are committed and these changes are reflected in the later phases.
- **Simple:** Iterative waterfall model is very simple to understand and use. That's why it is one of the most widely used software development models.

### **Drawbacks of Iterative Waterfall Model**

- **Difficult to incorporate change requests:** The major drawback of the iterative waterfall model is that all the requirements must be clearly stated before starting of the development phase. Customer may change requirements after some time but the iterative waterfall model does not leave any scope to incorporate change requests that are made after development phase starts.
- **Incremental delivery not supported:** In the iterative waterfall model, the full software is completely developed and tested before delivery to the customer. There is no scope for any intermediate delivery. So, customers have to wait long for getting the software.
- **Overlapping of phases not supported:** Iterative waterfall model assumes that one phase can start after completion of the previous phase, but in real projects, phases may overlap to reduce the effort and time needed to complete the project.
- **Risk handling not supported:** Projects may suffer from various types of risks. But Iterative waterfall model has no mechanism for risk handling.

- **Limited customer interactions:** Customer interaction occurs at the start of the project at the time of requirement gathering and at project completion at the time of software delivery. These fewer interactions with the customers may lead to many problems as the finally developed software may differ from the customers' actual requirements.

## **System implementation:**

### **Software Implementation Challenges:**

There are some challenges faced by the development team while implementing the software. Some of them are mentioned below:

- **Code-reuse** - Programming interfaces of present-day languages are very sophisticated and are equipped huge library functions. Still, to bring the cost down of end product, the organization management prefers to re-use the code, which was created earlier for some other software. There are huge issues faced by programmers for compatibility checks and deciding how much code to re-use.
- **Version Management** - Every time a new software is issued to the customer, developers have to maintain version and configuration related documentation. This documentation needs to be highly accurate and available on time.
- **Target-Host** - The software program, which is being developed in the organization, needs to be designed for host machines at the customers end. But at times, it is impossible to design a software that works on the target machines.

### **Structured Programming**

In the process of coding, the lines of code keep multiplying, thus, size of the software increases. Gradually, it becomes next to impossible to remember the flow of program. If one forgets how software and its underlying programs, files, procedures are constructed it then becomes very difficult to share, debug and modify the program. The solution to this is structured programming. It encourages the developer to use subroutines and loops instead of using simple jumps in the code, thereby bringing clarity in the code and improving its efficiency. Structured programming also helps programmer to reduce coding time and organize code properly.

Structured programming states how the program shall be coded. Structured programming uses three main concepts:



- **Top-down analysis** - A software is always made to perform some rational work. This rational work is known as problem in the software parlance. Hence it is very important that we understand how to solve the problem. Under top-down analysis, the problem is broken down into small pieces where each one has some significance. Each problem is individually solved and steps are clearly stated about how to solve the problem.
- **Modular Programming** - While programming, the code is broken down into smaller group of instructions. These groups are known as modules, subprograms or subroutines. Modular programming based on the understanding of top-down analysis. It discourages jumps using 'goto' statements in the program, which often makes the program flow non-traceable. Jumps are prohibited and modular format is encouraged in structured programming.
- **Structured Coding** - In reference with top-down analysis, structured coding sub-divides the modules into further smaller units of code in the order of their execution. Structured programming uses control structure, which controls the flow of the program, whereas structured coding uses control structure to organize its instructions in definable patterns.

## **Functional Programming**

Functional programming is style of programming language, which uses the concepts of mathematical functions. A function in mathematics should always produce the same result on receiving the same argument. In procedural languages, the flow of the program runs through procedures, i.e., the control of program is transferred to the called procedure. While control flow is transferring from one procedure to another, the program changes its state.

In procedural programming, it is possible for a procedure to produce different results when it is called with the same argument, as the program itself can be in different state while calling it. This is a property as well as a drawback of procedural programming, in which the sequence or timing of the procedure execution becomes important.

Functional programming provides means of computation as mathematical functions, which produces results irrespective of program state. This makes it possible to predict the behavior of the program.

Functional programming uses the following concepts:

- **First class and High-order functions** - These functions have capability to accept another function as argument or they return other functions as results.
- **Pure functions** - These functions do not include destructive updates, that is, they do not affect any I/O or memory and if they are not in use, they can easily be removed without hampering the rest of the program.

- **Recursion** - Recursion is a programming technique where a function calls itself and repeats the program code in it unless some pre-defined condition matches. Recursion is the way of creating loops in functional programming.
- **Strict evaluation** - It is a method of evaluating the expression passed to a function as an argument. Functional programming has two types of evaluation methods, strict (eager) or non-strict (lazy). Strict evaluation always evaluates the expression before invoking the function. Non-strict evaluation does not evaluate the expression unless it is needed.
- **$\lambda$ -calculus** - Most functional programming languages use  $\lambda$ -calculus as their type systems.  $\lambda$ -expressions are executed by evaluating them as they occur.

Common Lisp, Scala, Haskell, Erlang and F# are some examples of functional programming languages.

## **Programming style**

Programming style is set of coding rules followed by all the programmers to write the code. When multiple programmers work on the same software project, they frequently need to work with the program code written by some other developer. This becomes tedious or at times impossible, if all developers do not follow some standard programming style to code the program.

An appropriate programming style includes using function and variable names relevant to the intended task, using well-placed indentation, commenting code for the convenience of reader and overall presentation of code. This makes the program code readable and understandable by all, which in turn makes debugging and error solving easier. Also, proper coding style helps ease the documentation and updating.

## **DATA FLOW DIAGRAM**


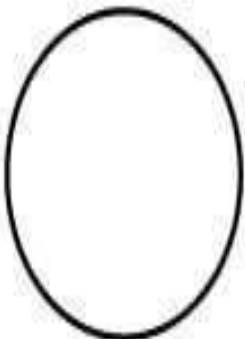

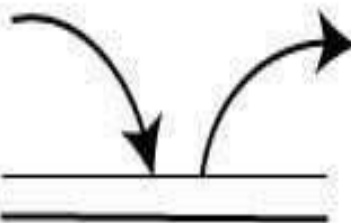
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

**The following observations about DFDs are essential:**

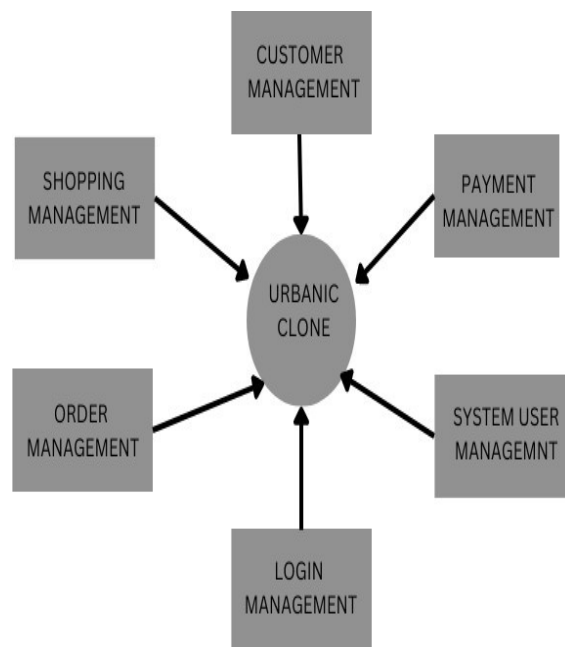
1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows in a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represent decision points with multiple existing paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Symbol	Name	Function
	Data flow	Used to Connect Processes to each other, to sources or Sinks; the arrow head indicates direction of data flow.
	Process	Performs Some transformation of Input data to yield output data.
	Source of Sink (External Entity)	A Source of System inputs or Sink of System outputs.
	Data Store	A repository of data; the arrow heads indicate net inputs and net outputs to store.

**Symbols for Data Flow Diagrams**

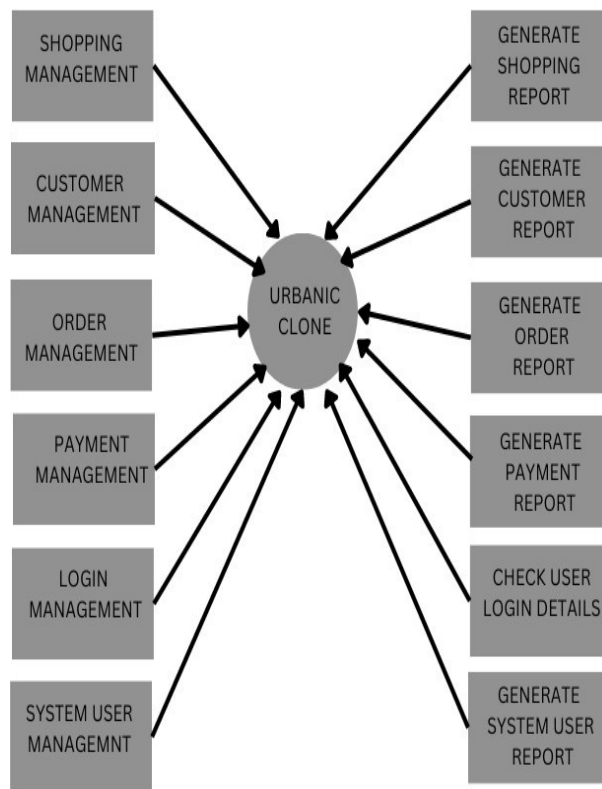
### **Level 0 DFD:**

This is the highest-level DFD, which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes.

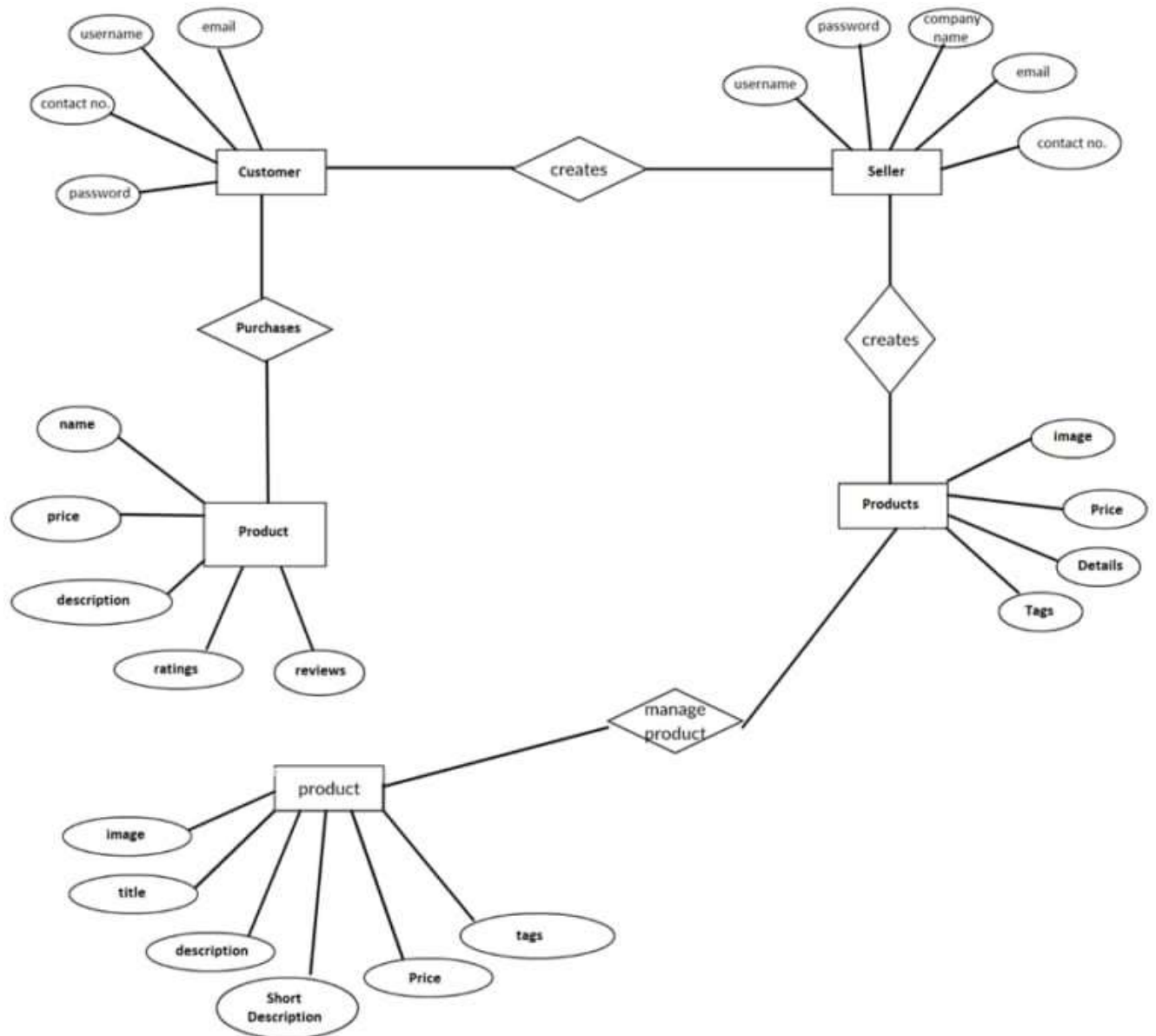


## **Level 1 DFD:**

This level provides a more detailed view of the system by breaking down the major processes identified in the level 0 DFD into sub-processes. Each sub-process is depicted as a separate process on the level 1 DFD. The data flows and data stores associated with each sub-process are also shown.



## ER DIAGRAM



Here are the geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section (Components of an ER Diagram) of this guide so don't worry too much about these terms now, just go through them once.

**Rectangle:** Represents Entity sets.

**Ellipses:** Attributes

**Diamonds:** Relationship Set

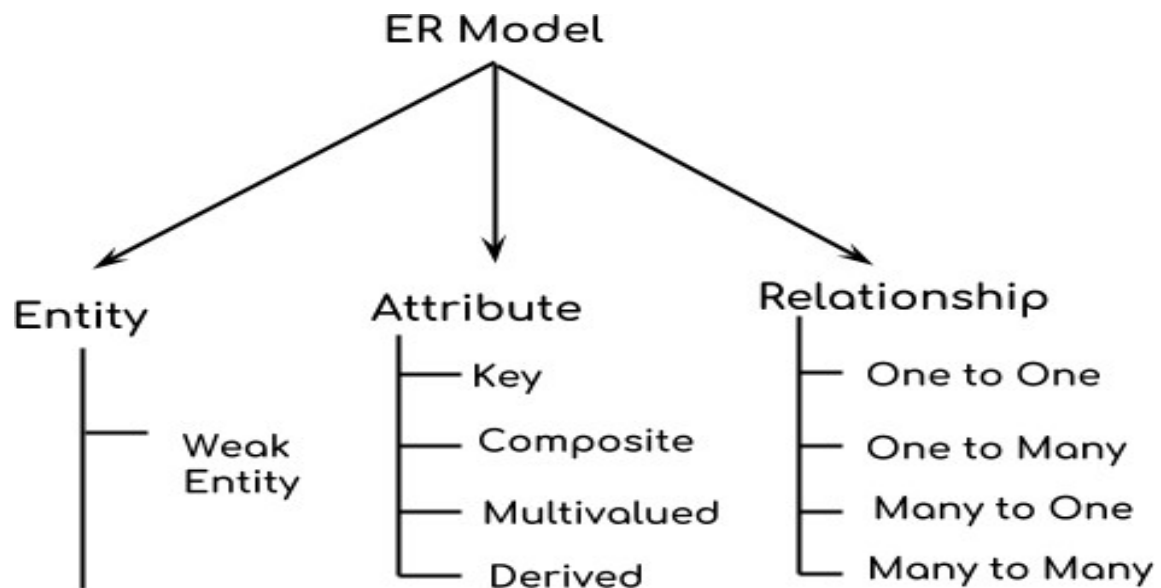
**Lines:** They link attributes to Entity Sets and Entity sets to Relationship Set

**Double Ellipses:** Multivalued Attributes

**Dashed Ellipses:** Derived Attributes

**Double Rectangles:** Weak Entity Sets

**Double Lines:** Total participation of an entity in a relationship set



### Components of ER Diagram



# **CHAPTER-4**

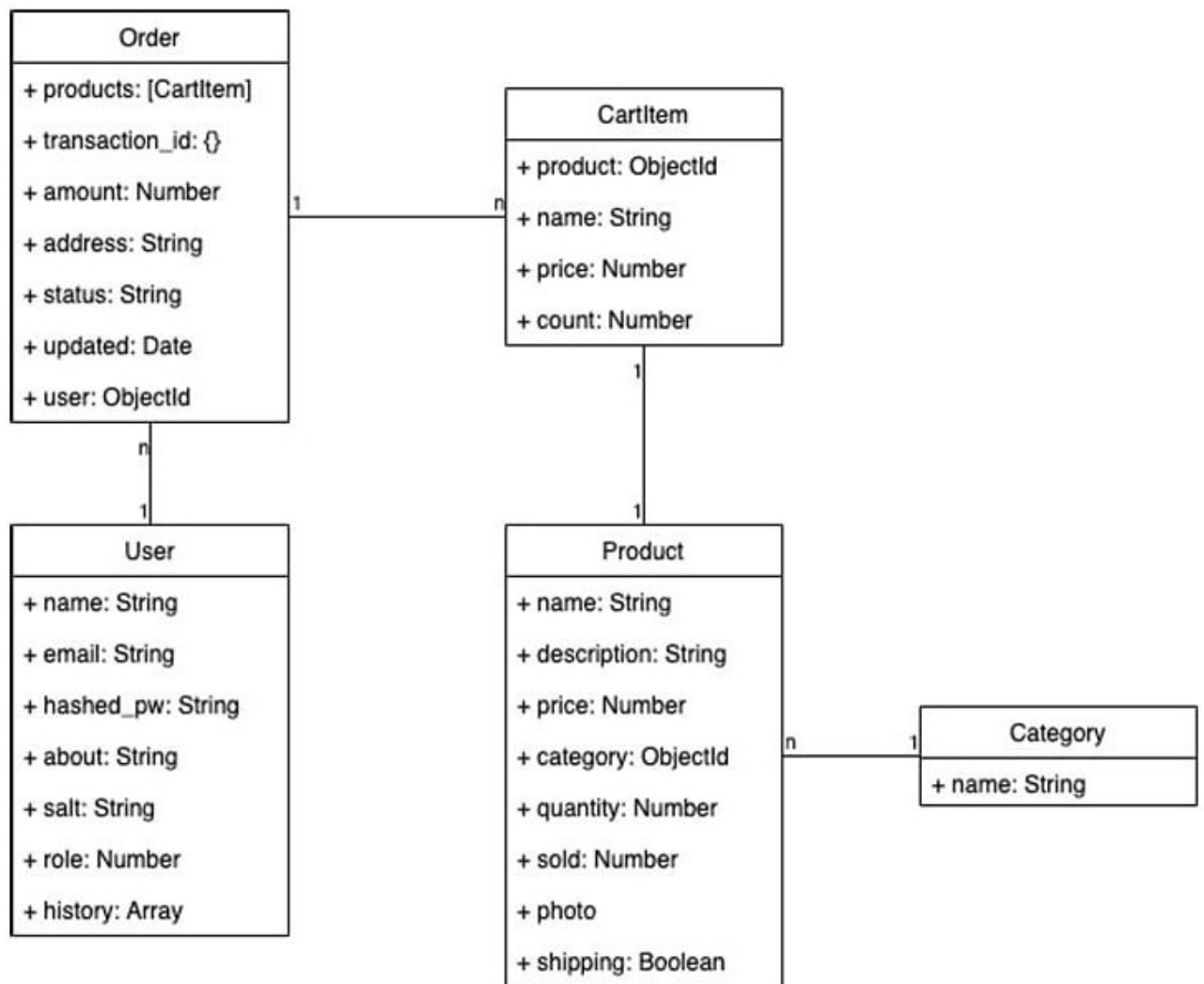
## **DATABASE SCHEMA**

A database schema is a blueprint that outlines the structure of a database, including tables, fields, relationships, and constraints. It defines the logical organization of data in a database and provides a framework for storing, retrieving, and managing data.

The database schema consists of several components:

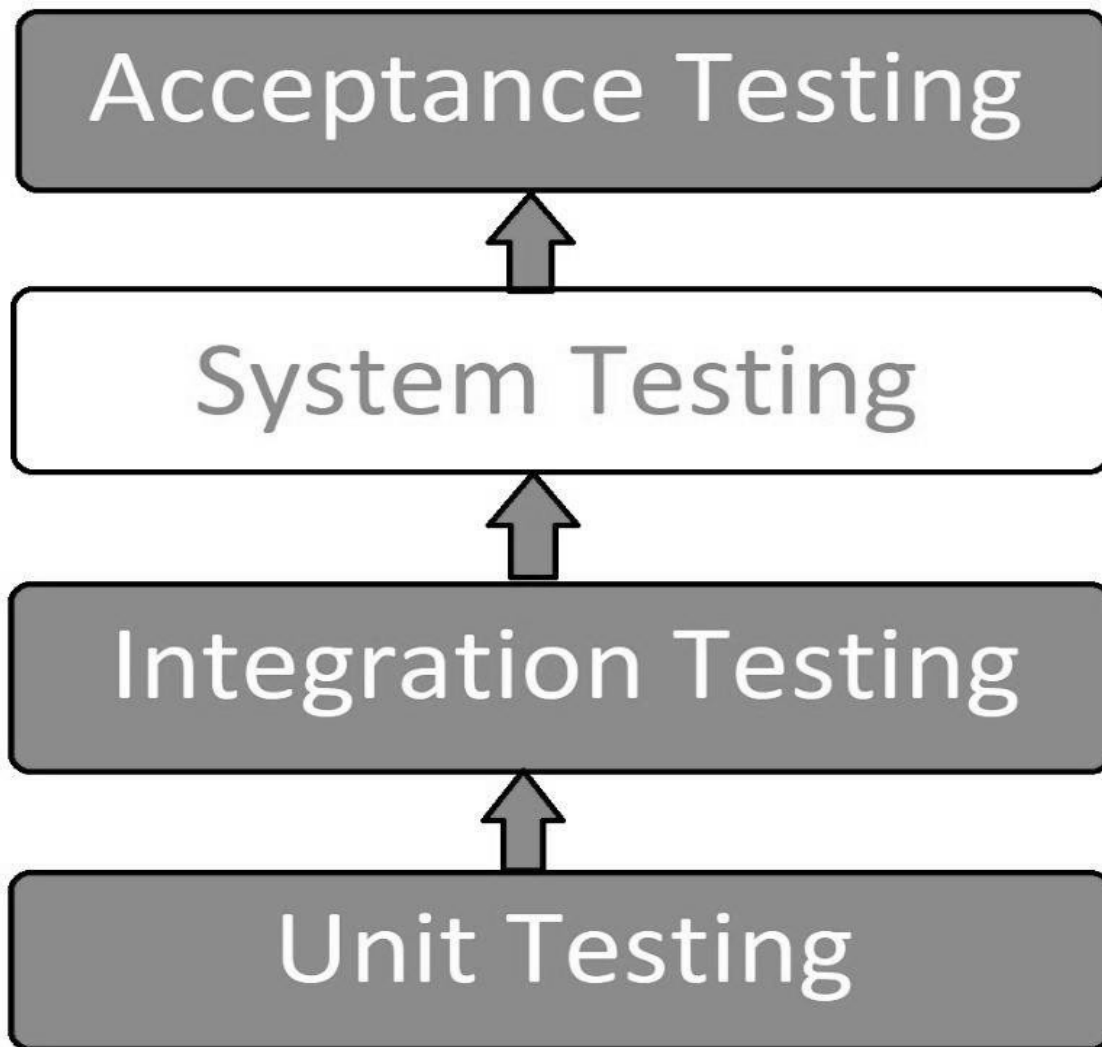
1. **Tables:** A table is a collection of related data organized into rows and columns. Tables represent entities or objects, such as customers, products, orders, or transactions.
2. **Fields:** A field is a column in a table that stores a specific type of data, such as text, numbers, dates, or images. Fields are also known as attributes or columns.
3. **Relationships:** A relationship defines how data in one table is related to data in another table. There are different types of relationships, such as one-to-one, one-to-many, or many-to-many.
4. **Primary keys:** A primary key is a unique identifier for each record in a table. It is used to ensure that each record is unique and to establish relationships between tables.
5. **Foreign keys:** A foreign key is a column in a table that refers to the primary key of another table. It is used to establish relationships between tables and ensure referential integrity.
6. **Constraints:** Constraints are rules that restrict the type or range of data that can be stored in a table. They can ensure data accuracy, consistency, and integrity.

The database schema is an essential component of a database management system (DBMS). It provides a standardized way of organizing data and facilitates efficient data access and management. The schema can be designed using various tools, such as ER diagrams, diagrams, or data modeling software. It should be designed carefully to ensure that it meets the requirements of the application and can scale as needed.



## **SYSTEM TESTING**

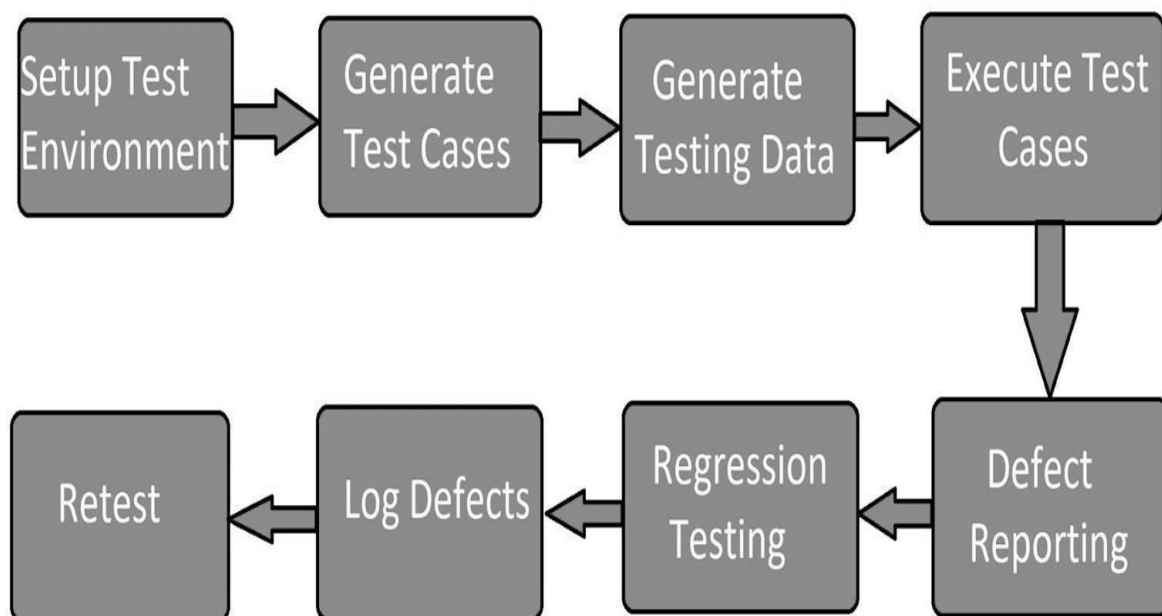
System Testing is performed after the integration testing and before the acceptance testing.



## System Testing Process:

System Testing is performed in the following steps:

- **Test Environment Setup:**  
Create testing environment for the better-quality testing.
- **Create Test Case:**  
Generate test case for the testing process.
- **Create Test Data:**  
Generate the data that is to be tested.
- **Execute Test Case:**  
After the generation of the test case and the test data, test cases are executed.
- **Defect Reporting:**  
Defects in the system are detected.
- **Regression Testing:**  
It is carried out to test the side effects of the testing process.
- **Log Defects:**  
Defects are fixed in this step.
- **Retest:**  
If the test is not successful then again test is performed.



## **Types of System Testing:**

- **Performance Testing:**

Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability and reliability of the software product or application.

- **Load Testing:**

Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.

- **Stress Testing:**

Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.

- **Scalability Testing:**

Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load.

## **Software and Hardware Testing**

This testing of the system intends to check hardware and software compatibility. The hardware configuration must be compatible with the software to run it without any issue. Compatibility provides flexibility by providing interactions between hardware and software.

Why is System Testing Important?

- System Testing gives hundred percent assurance of system performance as it covers end to end function of the system.
- It includes testing of System software architecture and business requirements.
- It helps in mitigating live issues and bugs even after production.
- System testing uses both existing system and a new system to feed same data in both and then compare the differences in functionalities of added and existing functions so, the user can understand benefits of new added functions of the system.

### **Performance testing process:**

The performance testing cannot be done manually since:

- We need a lot of resources, and it became a costlier approach.
- And the accuracy cannot maintain when we track response time manually.

The Performance testing process will be completed in the following steps:

- Identify performance scenarios
- Plan and design performance test script
- Configure the test environment & distribute the load
- Execute test scripts
- Result
- Analysis result
- Identify the Bottleneck
- Re-run test



If we perform a **positive flow** of the performance testing process, it could follow the below process:

### **Identify performance scenarios**

Firstly, we will identify the performance scenarios based on these below factors:

**Most commonly scenarios:** It means that we can find the performance scenarios based on the scenarios, which commonly used like in the **Gmail application**; we will perform **login, inbox, send items, and compose a mail and logout**.

**Most critical scenarios:** Critical scenarios mean regularly used and important for the business-like in Gmail application **login, compose, inbox, and logout**.

**Huge data transaction:** If we have huge data means that n-number of the users using the application at the same time.

Once we identify the performance scenarios, we will move to the next step.



### **Plan and design performance test script**

In this step, we will install the tools in the Test Engineer Machine and access the test server and then we write some script according to the test scenarios and run the tool.

Once we are done with writing the script, we will go to the next step.

### **Configure the test environment & distribute the load**

After writing the test scripts, we will arrange the testing environment before the execution. And also, manage the tools, other resources and distribute the load according to the "Usage Pattern" or mention the duration and stability.

### **Execute test scripts**

Once we are done with distributing the load, we will execute, validate, and monitor the test scripts.

### **Result**

After executing the test scripts, we will get the test result. And check that the result meeting the goal in the given response time or not, and the response time could be maximum, average, and minimum.

If the response is not meeting the required time response, then we will go for the **negative flow** where will perform the below steps:

### **Analysis result**

First, we will analyze the test result whether it meets with the response time or not.

### **Identify the Bottleneck**

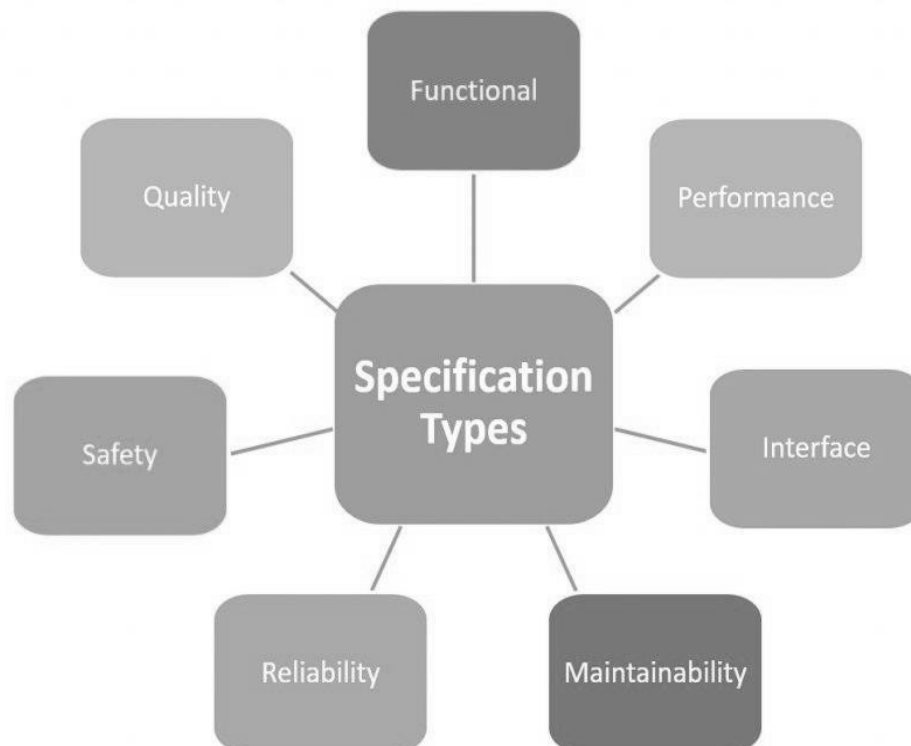
After that, we will identify the **bottleneck (bug or performance issue)**. And the bottleneck could occur because of these aspects like the **problem in code, hardware issue (hard disk,**

**RAM Processor), network issues, and the software issue (operating system).** And after finding the bottleneck, we will perform **tuning (fix or adjustment)** to resolve this bottleneck.

### **Re-run test**

Once we fix the bottlenecks, re-run the test scripts and checks the result whether it meets the required goal or not.

## Software Requirement Specifications



A **software requirements specification (SRS)** is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based

the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system, we should have clear understanding of Software system. To achieve that we need to continuous communication with customers to gather all requirements.

A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real-life scenarios. Using the *Software requirements specification* (SRS) document on QA lead, managers create test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results.

It is highly recommended to review or test SRS documents before start writing test cases and making any plan for testing. Let's see how to test SRS and the important point to keep in mind while testing it.

**1. Correctness of SRS should be checked:** Since the whole testing phase is dependent on SRS, it is very important to check its correctness. There are some standards with which we can compare and verify.

**2. Ambiguity should be avoided:** Sometimes in SRS, some words have more than one meaning and this might confuse testers making it difficult to get the exact reference. It is advisable to check for such ambiguous words and make the meaning clear for better understanding.

**3. Requirements should be complete:** When tester writes test cases, what exactly is required from the application, is the first thing which needs to be clear. For e.g.: if application needs to send the specific data of some specific size, then it should be clearly mentioned in SRS that how much data and what is the size limit to send.

**4. Consistent requirements:** The SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another. This sets the standard and should be followed throughout the testing phase.

**5. Verification of expected result:** SRS should not have statements like "Work as expected", it should be clearly stated that what is expected since different testers would have different thinking aspects and may draw different results from this statement.

**6. Testing environment:** Some applications need specific conditions to test and also a particular environment for accurate result. SRS should have clear documentation on what type of environment is needed to set up.

**7. Pre-conditions defined clearly:** One of the most important parts of test cases is pre-conditions. If they are not met properly then actual result will always be different expected result. Verify that in SRS, all the pre-conditions are mentioned clearly.

**8. Requirements ID:** These are the base of test case template. Based on requirement Ids, test case ids are written. Also, requirements ids make it easy to categorize modules so just by looking at them, tester will know which module to refer. SRS must have them such as id defines a particular module.

**9. Security and Performance criteria:** Security is priority when a software is tested especially when it is built in such a way that it contains some crucial information when leaked can cause harm to business. Tester should check that all the security related requirements are properly defined and are clear to him. Also, when we talk about performance of a software, it plays a very important role in business so all the requirements related to performance must be clear to the tester and he must also know when and how much stress or load testing should be done to test the performance.

**10. Assumption should be avoided:** Sometimes when requirement is not cleared to tester, he tends to make some assumptions related to it, which is not a right way to do testing as assumptions could go wrong and hence, test results may vary. It is better to avoid assumptions and ask clients about all the “missing requirements” to have a better understanding of expected results.

**11. Deletion of irrelevant requirements:** There are more than one team who work on SRS so it might be possible that some irrelevant requirements are included in SRS. Based on the understanding of the software, tester can find out which are these requirements and remove them to avoid confusions and reduce work load.

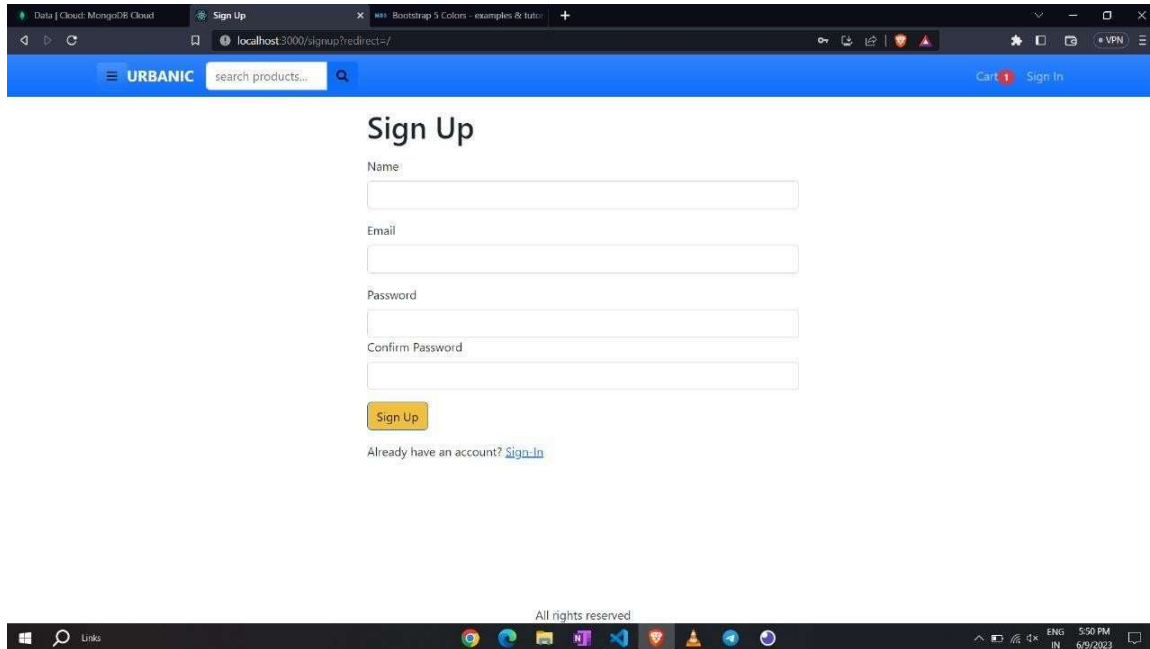
**12. Freeze requirements:** When an ambiguous or incomplete requirement is sent to client to analyze and tester gets a reply, that requirement result will be updated in the next SRS version and client will freeze that requirement. Freezing here means that result will not change again until and unless some major addition or modification is introduced in the software.

Most of the defects which we find during testing are because of either incomplete requirements or ambiguity in SRS. To avoid such type of defects it is very important to test software requirements specification before writing the test cases. Keep the latest version of SRS with you for reference and keep yourself updated with the latest change made to the SRS. Best practice is

to go through the document very carefully and note down all the confusions, assumptions and incomplete requirements and then have a meeting with the client to get them clear before development phase starts as it becomes costly to fix the bugs after the software is developed. After all the requirements are cleared to a tester, it becomes easy for him to write effective test cases and accurate expected results.

Over to you: I think I have addressed all major points of Software requirements specification. Have you ever worked on “**Testing of software requirements specification (SRS) document**”. If yes then please make sure that you share it with your QA friends. Please leave your questions/tips/suggestions in the comment section below and I’ll try to answer as many as I can.

## SCREENSHOTS



The screenshot shows a web browser window with the URL `localhost:3000/signup?redirect=`. The page has a blue header with the "URBANIC" logo and a search bar. The main content area is titled "Sign Up" and contains a form with the following fields: Name, Email, Password, and Confirm Password. A yellow "Sign Up" button is located below the form. A link "Already have an account? [Sign In](#)" is at the bottom. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system tray on the right indicates the language is "ENG IN" and the date is "6/9/2023".

Sign Up

Name

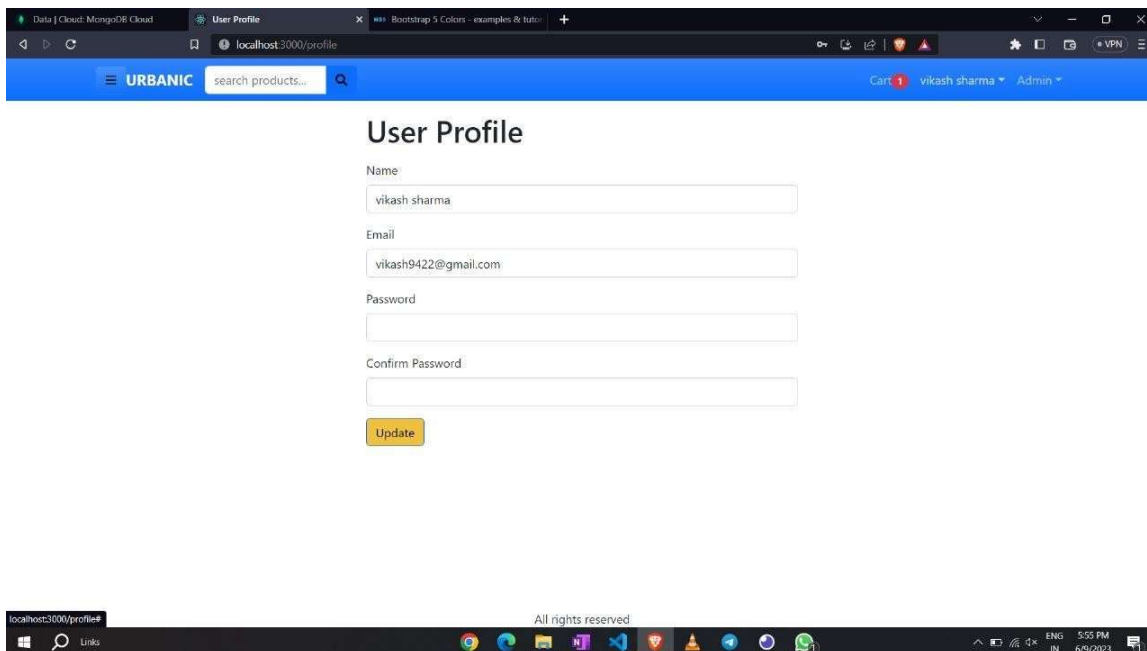
Email

Password

Confirm Password

Sign Up

Already have an account? [Sign In](#)



The screenshot shows a web browser window with the URL `localhost:3000/profile`. The page has a blue header with the "URBANIC" logo and a search bar. The main content area is titled "User Profile" and contains a form with the following fields: Name, Email, Password, and Confirm Password. The "Name" field is pre-filled with "vikash sharma" and the "Email" field is pre-filled with "vikash9422@gmail.com". A yellow "Update" button is located below the form. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system tray on the right indicates the language is "ENG IN" and the date is "6/9/2023".

User Profile

Name

vikash sharma

Email

vikash9422@gmail.com

Password

Confirm Password

Update

localhost:3000/admin/Users#

Users

ID	NAME	EMAIL	IS ADMIN	ACTIONS
6482ea4748e9d7d4676a5b7f	vikash sharma	vikash9422@gmail.com	YES	<a href="#">Edit</a> <a href="#">Delete</a>

localhost:3000/admin/Users# All rights reserved

localhost:3000/profile#

User Profile

Name

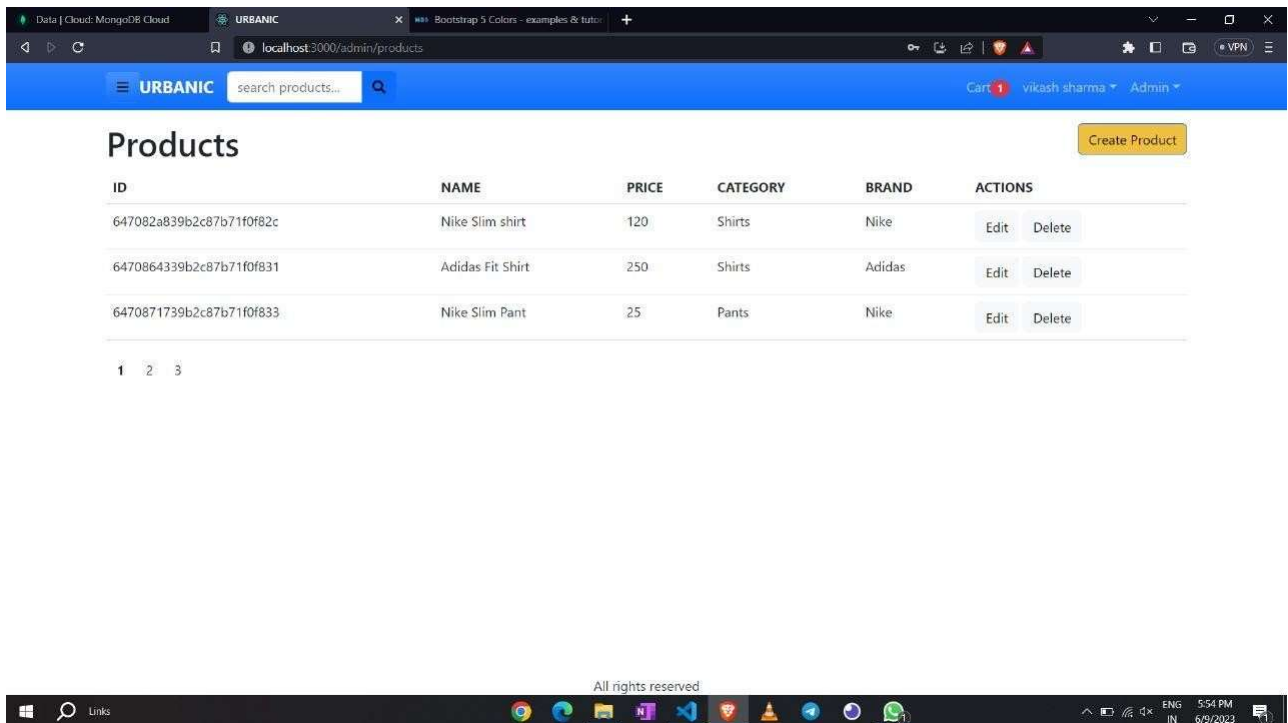
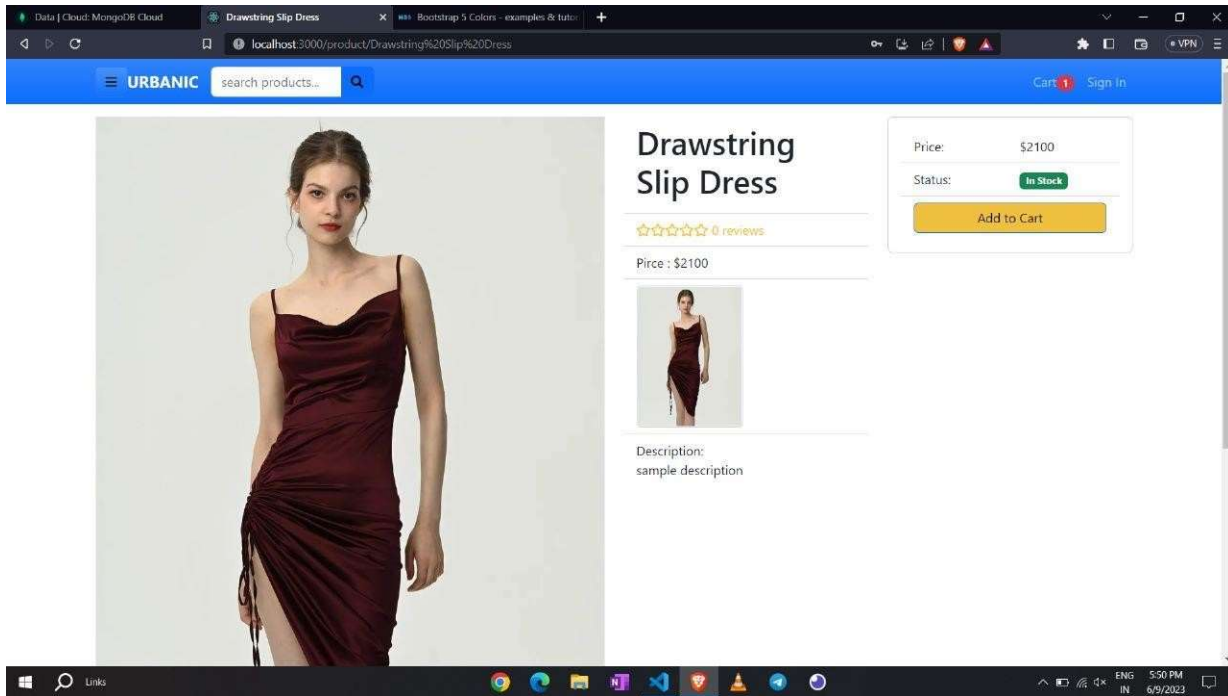
Email

Password

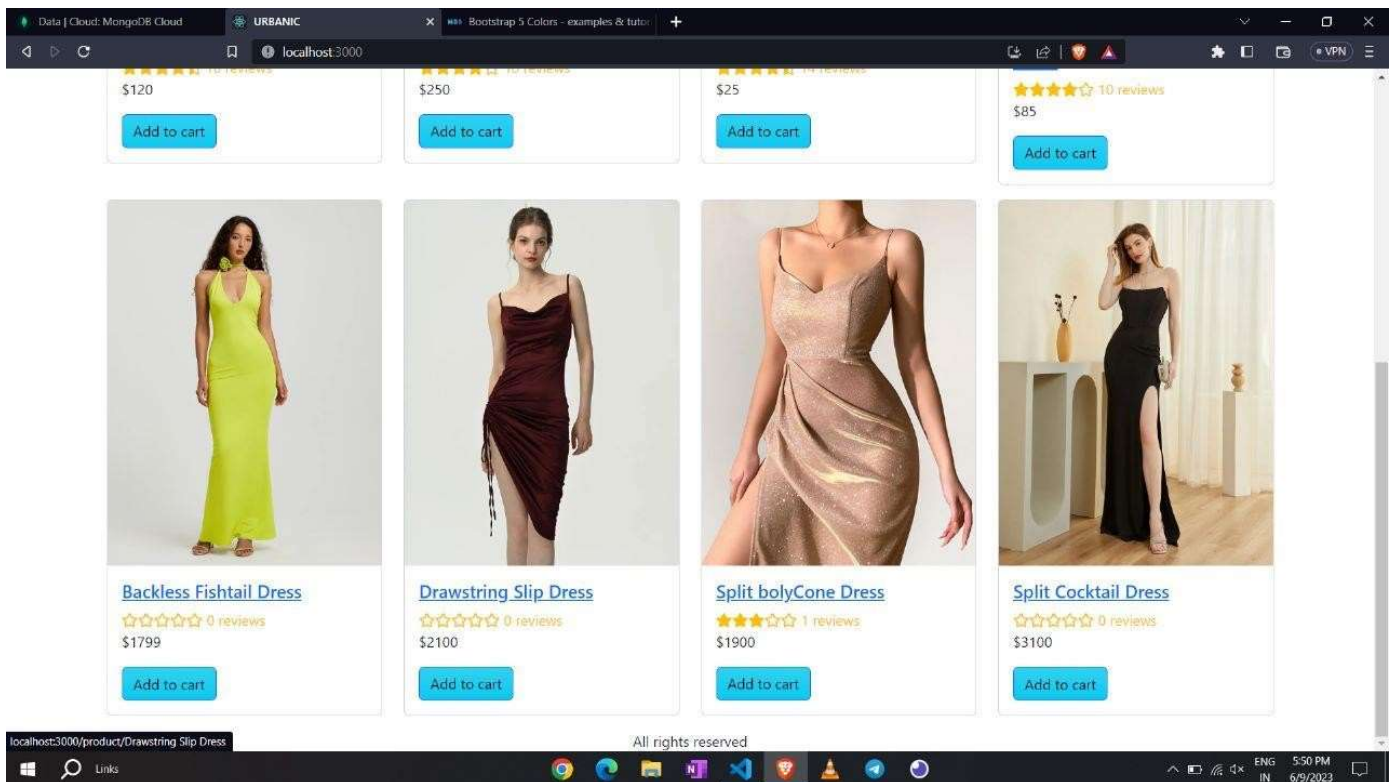
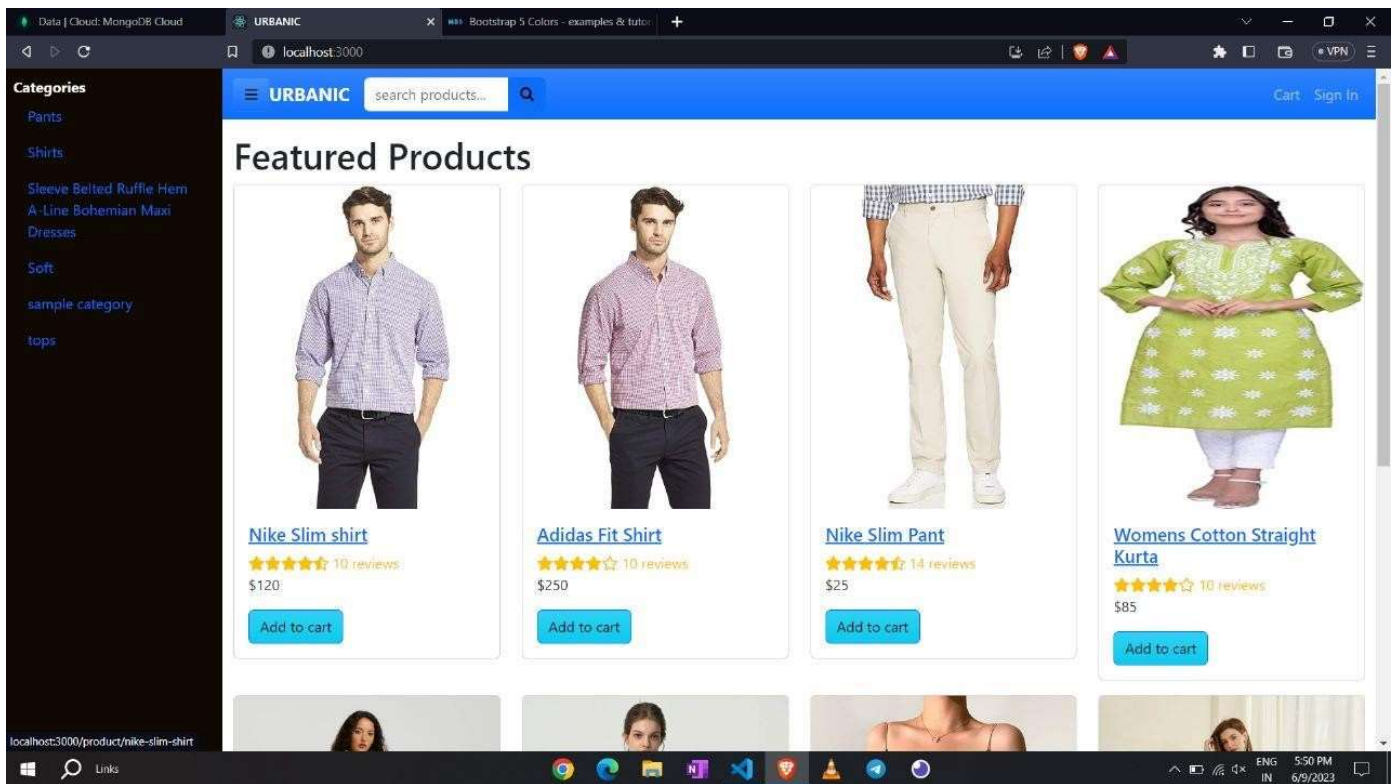
Confirm Password

[Update](#)

localhost:3000/profile# All rights reserved







## Products

Create Product

ID	NAME	PRICE	CATEGORY	BRAND	ACTIONS
647082a839b2c87b71f0f82c	Nike Slim shirt	120	Shirts	Nike	Edit Delete
6470864339b2c87b71f0f831	Adidas Fit Shirt	250	Shirts	Adidas	Edit Delete
6470871739b2c87b71f0f833	Nike Slim Pant	25	Pants	Nike	Edit Delete

1 2 3

## Orders

ID	USER	DATE	TOTAL	PAID	DELIVERED	ACTIONS
----	------	------	-------	------	-----------	---------

## **CODING:**

### **DATABASE CONNECTIVITY**

```
> frontend
) lessons
.gitignore
O README_md
> TIMELINE

slug: { type: String, required: true
name: { type: String, required: true } ,
quantity: { type: Number, required: true } ,
image: type: String, required: true } ,
price: type: Number, required: true } ,
product :
type: mongoose . Types
ref: Product' ,
required: true,
shippingAddress:
fuUName; type: String, required: true
address: { type: String, required: true } ,
city: type: String, required: true } ,
postalCode: type: String, required: true } ,
country: { type: String, required: true } ,
Location: {
lat: Number,
Ing: Number,
address: String,
```

```

name: String,
vicinity: String,
googleAddressId: String,
type: String,
paymentMethod :
payment Result:
id: String,
status: String,
required:

```

```
data_js
```

```
{9
```

```
Edit Selection View GO
```

```
- URBANIC
```

```
App.js
```

```
frontend > src App.is >
```

```
import Resetpasswordscreen from ./screens/ResetPasswordScreen•;
```

```
f unctuon App()
```

```
const State, dispatch: CtxDispatCh
```

```
useContext(Store);
```

```
const ( full180x, cart, userInfo =
```

```
state;
```

```
CtXDispatCh(4 type: 1);
```

```
tocalstorage. removertem( •user Info');
```

```
LocalStorage-r
```

```
«vertem( shippingAddress• j;
```

```
localStorage. remvertem( ) ;
```

```
slug: { type: String, required: true
```

```

name: { type: String, required: true } ,
quantity: { type: Number, required: true } ,
image: type: String, required: true } ,
price: type: Number, required: true } ,
product :
type: mongoose . Types
ref: Product' ,
required: true,
shippingAddress:
fuUName; type: String, required: true
address: { type: String, required: true } ,
city: type: String, required: true } ,
postalCode: type: String, required: true } ,
country: { type: String, required: true } ,
E ndex.JS
logo-svg
reporWebVitaIs.js
setup Tests-js
Sto JS
package-lockjson
@ package.json
O READMEmd
lessons
O README.rnd
> TIMELINE
import
import
import

```

```

import
  React from 'react'
  { createRoot from 'react-dom/client' }
  { HelmetProvider from 'react-helmet-async' }
  { PayPalScriptProvider } from '@paypal/react-paypal-script'
  'bootstrap/dist/css/bootstrap.min.css'
  './index.css';

App from './App';

reportWebVitals from './reportWebVitals';

{ StoreProvider from './store' }
'root' ));

const root = createRoot(
  <React.StrictMode>
    <HelmetProvider>
      <PayPalScriptProvider>
        <App />
      <PayPalScriptProvider>
        <StoreProvider>
          <React.StrictMode>
            // If you want to start measuring performance in your app,
            // pass a function
            // to log results (for example: reportWebVitals(console.log))
            // or send to an analytics endpoint.
            reportWebVitals() ;

            Learn more: https://bit.ly/CRA-vitals

            slug: { type: String, required: true

```

```

name: { type: String, required: true } ,
quantity: { type: Number, required: true } ,
image: type: String, required: true } ,
price: type: Number, required: true } ,
product :
type: mongoose . Types
ref: Product' ,
required: true,
shippingAddress:
fuUName; type: String, required: true
address: { type: String, required: true } ,
city: type: String, required: true } ,
postalCode: type: String, required: true } ,
country: { type: String, required: true } ,
err.'
data. js
(i) package-lockjson
> frontend
) lessons
.gitignore
O README_md
> TIMELINE
slug: { type: String, required: true
name: { type: String, required: true } ,
quantity: { type: Number, required: true } ,
image: type: String, required: true } ,
price: type: Number, required: true } ,
product :
type: mongoose . Types

```

```
ref: Product' ,
required: true,
shippingAddress:
fuUName; type; String, required; true
address: { type: String, required: true } ,
city: type: String, required: true } ,
postalCode: type: String, required; true } ,
country: { type: String, required: true } ,
Location: {
lat: Number,
Ing: Number,
address: String,
name: String,
vicinity: String,
```



## **SOFTWARE MAINTENANCE**

Software maintenance is widely accepted part of SDLC now a days. It stands for all the modifications and updations done after the delivery of software product. There are number of reasons, why modifications are required, some of them are briefly mentioned below:

- **Market Conditions** - Policies, which changes over the time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger need for modification.
- **Client Requirements** - Over the time, customer may ask for new features or functions in the software.
- **Host Modifications** - If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.
- **Organization Changes** - If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.

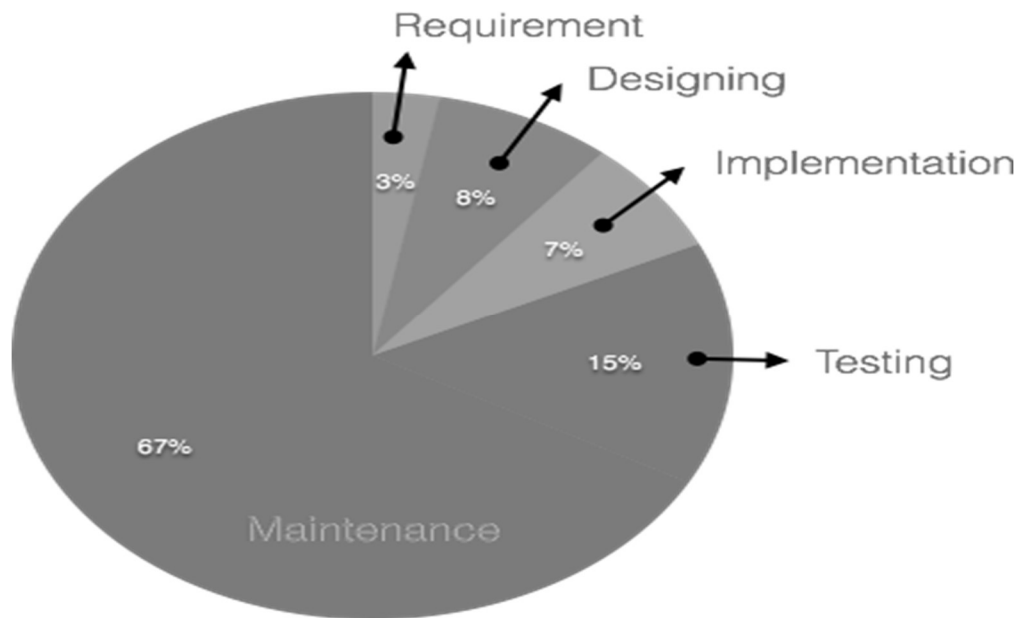
### **Types of maintenance**

In a software lifetime, type of maintenance may vary based on its nature. It may be just a routine maintenance tasks as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature. Following are some types of maintenance based on their characteristics:

- **Corrective Maintenance** - This includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.
- **Adaptive Maintenance** - This includes modifications and updations applied to keep the software product up-to date and tuned to the ever-changing world of technology and business environment.
- **Perfective Maintenance** - This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.
- **Preventive Maintenance** - This includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

## **Cost of Maintenance**

Reports suggest that the cost of maintenance is high. A study on estimating software maintenance found that the cost of maintenance is as high as 67% of the cost of entire software process cycle.



On an average, the cost of software maintenance is more than 50% of all SDLC phases.

## **CONCLUSION**

In conclusion, an Urbanic clone project can be a powerful e-commerce platform for users to browse, purchase, and manage their orders. Through the use of modern web development technologies and a robust database, such a project can provide a seamless user experience and support a large number of customers.

The proposed system for an Urbanic clone project aims to address the limitations of the existing system and provide a more robust and scalable platform for e-commerce. By implementing modules such as User Management, Product Management, Admin Management, and Database Connectivity, the project can offer a comprehensive set of features and capabilities to both customers and administrators.

With a well-designed architecture, appropriate software development life cycle, and efficient database schema design, the Urbanic clone project can be implemented effectively to meet the project's objectives. The use of Python as the primary programming language offers a powerful and flexible toolset for developing the application's core functionality.

Overall, an Urbanic clone project can be an exciting and rewarding project for developers looking to build a high-quality e-commerce platform. With careful planning, design, and implementation, the project can provide a valuable service to customers and help drive business growth for the organization.

## **FUTURE SCOPE**

The future scope of a scholarship portal is vast and presents opportunities for educational institutions and students to get connected and access more educational funding. Here are some of the future scopes of a scholarship portal:

- Scholarship portals will become more global, eventually connecting students from different parts of the world to prestigious universities globally.
- With advancements in technology, scholarship portals will explore personalization technology to enable students and institutions to match their varied criteria. It will provide an opportunity for educational institutions to develop personalized scholarship programs that suit their specific needs.
- Future scholarship portals will need to explore algorithms that will process and determine the eligibility of students for scholarships, making the selection process more efficient and effective.

In conclusion, the future of the scholarship portal is bright, with more advanced technology, customized, and personalized solutions that will meet the various needs of students and institutions. These developments will improve the scholarship application process, increase student success and access to education, and eventually have a positive impact on the socio-economic status of countries.

## **REFERENCES**

- [https://www.youtube.com/playlist?list=PL0b6OzIxLPbyrzCMJOFzLnF\\_-5E\\_dkzs](https://www.youtube.com/playlist?list=PL0b6OzIxLPbyrzCMJOFzLnF_-5E_dkzs)
- <https://www.w3schools.com/php/>  
<https://phdtalks.org/2021/03/create-dynamic-website-using-php.html> Real World Software Development by Richard Warburton.
- <https://www.geeksforgeeks.org/three-tier-client-server-architecture-in-distributed-system/>
- HTML and CSS: Design and build websites -by **Jon Duckett**.
- <https://www.sih.gov.in/>
- [https://www.youtube.com/playlist?list=PL0b6OzIxLPbyrzCMJOFzLnF\\_-5E\\_dkzs](https://www.youtube.com/playlist?list=PL0b6OzIxLPbyrzCMJOFzLnF_-5E_dkzs)
- <https://www.w3schools.com/php/>
- <https://phdtalks.org/2021/03/create-dynamic-website-using-php.html>
- <https://dbdesigner.page.link/vU67BDwRBE4XHYab6>
- <https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/>
- <https://laravel.com/>
- <https://getbootstrap.com/>