

1.INTRODUCTION

1. INTRODUCTION

PROBLEM DEFINITION

The project titled “The Beauty Hive” is an endeavor to provide a solution to this, the users can easily access the system by registering themselves for the services .The system will provide information of product selling of beauty look and services .The users can also view the beauty store facility , services and hair stylists based on that they can book the suited beautician or hairstylists as per their desired time and date .The users can also purchase various beauty products that are available online.

This includes a variety of collections of different beauty products .The customers can fulfill their needs by prior appointment of services.

SCOPE OF THE PROJECT

- Types of beauty products and services can also be added to the system and could be updated or edited by the admin.
- The purchasing and request for beauty products and services is done by the user itself along with respective details.
- The method of payment is also available for customer to proceed with payments a The customer details are maintained by the admin.
- The delivery details of regarding the customer are also available.

Modules in the software

-
- LOGIN
 - CUSTOMER
 - PRODUCTS
 - PACKAGES
 - ORDER
 - BEAUTICIAN COURSE
 - BEAUTY TIPS
 - PAYMENT

LOGIN MODULE :It consists of username and password so that there is easy accessibility for the customer to log in and review the products and fulfill their needs .

CUSTOMER MODULE : It consists of details of customer such as the address, phone number ,name , so that the admin can easily fulfill the customer needs and it helps the admin to have proper information about the customer.

PRODUCTS MODULE : It consists the products of beauty look that will be easy for customer to select for themselves a particular suitable for the face.

PACKAGES : In this section admin can manage services of the beauty look .

ORDER MODULE: It is for ordering the beauty products which customer wants to purchase.

BEAUTICIAN COURSE TRAINING: The customer can select particular course which they want to do in beauty look.

BEAUTY TIPS : User sees the details of beauty products and services.

PAYMENT MODULE: It is to pay the cash for the purchased beauty products by the customer.

2.SYSTEM STUDY

2. SYSTEM STUDY

2.1 EXISTING SYSTEM :

The existing system is a manual system. The proposed system tries to simplify the difficulties encountered in manually handling the details of the item ordered, payment and fulfilling the needs of customer. In this case the details of orders and payments are stored in one or more registers.

The existing system needs number of staff and a lot of time consumption for the process. When we compare the existing system with the proposed system, there are many drawbacks for the existing system.

Some of the main problems in the existing system is:

The existing system needs a number of registers for handling the records. This leads to tedious manual labor.

The calculations require a large amount of clerical time.

The system is less reliable due to the possibility of human errors that occur while entering data and during calculations.

- Low accuracy.
- Updating of written records is very difficult.
- Data processing is very low.
- Accessing previously stored records is a time consuming task.

FEASIBILITY STUDY

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. A feasibility study tests the viability of an idea, a project or even a new business. The goal of a feasibility study is to place emphasis on potential problems that could occur if a project is pursued and determines if, after all significant factors are considered, the project should be pursued. Feasibility studies also allow a business to address where and how it will operate, potential obstacles, competition and the Funding needed to get the business up and running.

This project "THE BEAUTY HIVE" has undergone the following Feasibility study:

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility
- Schedule Feasibility

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Feasibility and risk analysis are related in many ways. The feasibility analysis in this project has been discussed below based on the above-mentioned components of feasibility.

Technical feasibility:

Technical feasibility centers on the technology used. It means the computerized system is technically feasible i.e., it doesn't have any technical fault and work properly in the given environment. Our system is technically feasible it is providing us the required output.

Economic feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of the computerized system. We analyze the computerized system is feasible as than the manual system because it saves the money, time and manpower. It is also feasible according to cost benefit analysis.

Behavioral feasibility:

Behavioral feasibility is the analysis of behavior of the computerized system. In this we analysis that the computerized system is working properly or not. If working then it is communicating properly with the environment or not. All the matters are analyzed and a good computerized system is prepared.

1. Schedule feasibility:

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems.

PROPOSED SYSTEM

The main objective of the system is to provide the users with accurate information and service contents for them to choose from, with complete transparency and by eliminating unwanted charges applied and a reliable temple online platform for easier booking with less time consumption.

- Easy booking
- Customer satisfaction.
- Efficient time consumption.
- Selection in method of payment.

3. SYSTEM DESIGN

3. SYSTEM DESIGN

In the design phase the architecture is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture. The architecture defines the components, their interfaces and behaviors. The deliverable design document is the architecture.

The design document describes a plan to implement the requirements. This phase represents the "how" phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established. The design may include the usage of existing components. Analyzing the trade-offs of necessary complexity allows for many things to remain simple which, in turn, will eventually lead to a higher quality product. The architecture team also converts the typical scenarios into a test plan.

In our approach, the team, given a complete requirement document, must also indicate critical priorities for the implementation team. A critical implementation priority leads to a task that has to be done right. If it fails, the product fails. If it succeeds, the product might succeed. At the very least, the confidence level of the team producing a successful product will increase. This will keep the implementation team focused. Exactly how this information is conveyed is a skill based on experience more than a science based on fundamental foundations.

System design is the process of defining the architecture components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

Until the 1990s, systems design had a crucial and respected role in the data processing industry. In the 1990s, standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

LOGICAL DESIGN:

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

PHYSICAL DESIGN:

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified /authenticated, how it is processed, and how it is displayed.

In physical design, the following requirements about the system are decided.

- Input requirement,
- Output requirements,
- Storage requirements,
- Processing requirements,
- System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. It is concerned with how the data is represented and stored within the system. Finally, **Process Design** is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system.

At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase. Physical design, in this context, does not refer to the tangible physical design of an information system.

To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and control processor. The H/S personal specification is developed for the proposed system.

E-R DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

Structure of an Entity Relationship Diagram with Common ERD Notations

An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

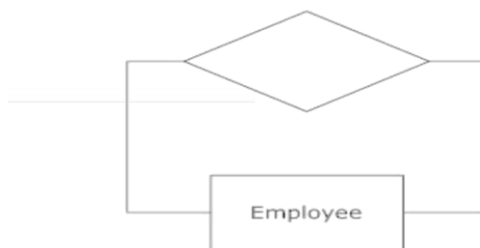
- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.



- **Weak entity** is an entity that must be defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



- **Actions**, which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.



- **Relationship**: The degree of a relationship is the number of entity types that participate in the relationship.



- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



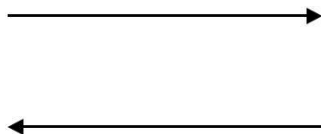
- **Multi-valued attribute** can have more than one value. For example, an employee entity can have multiple skill values.



- **Derived attribute** is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

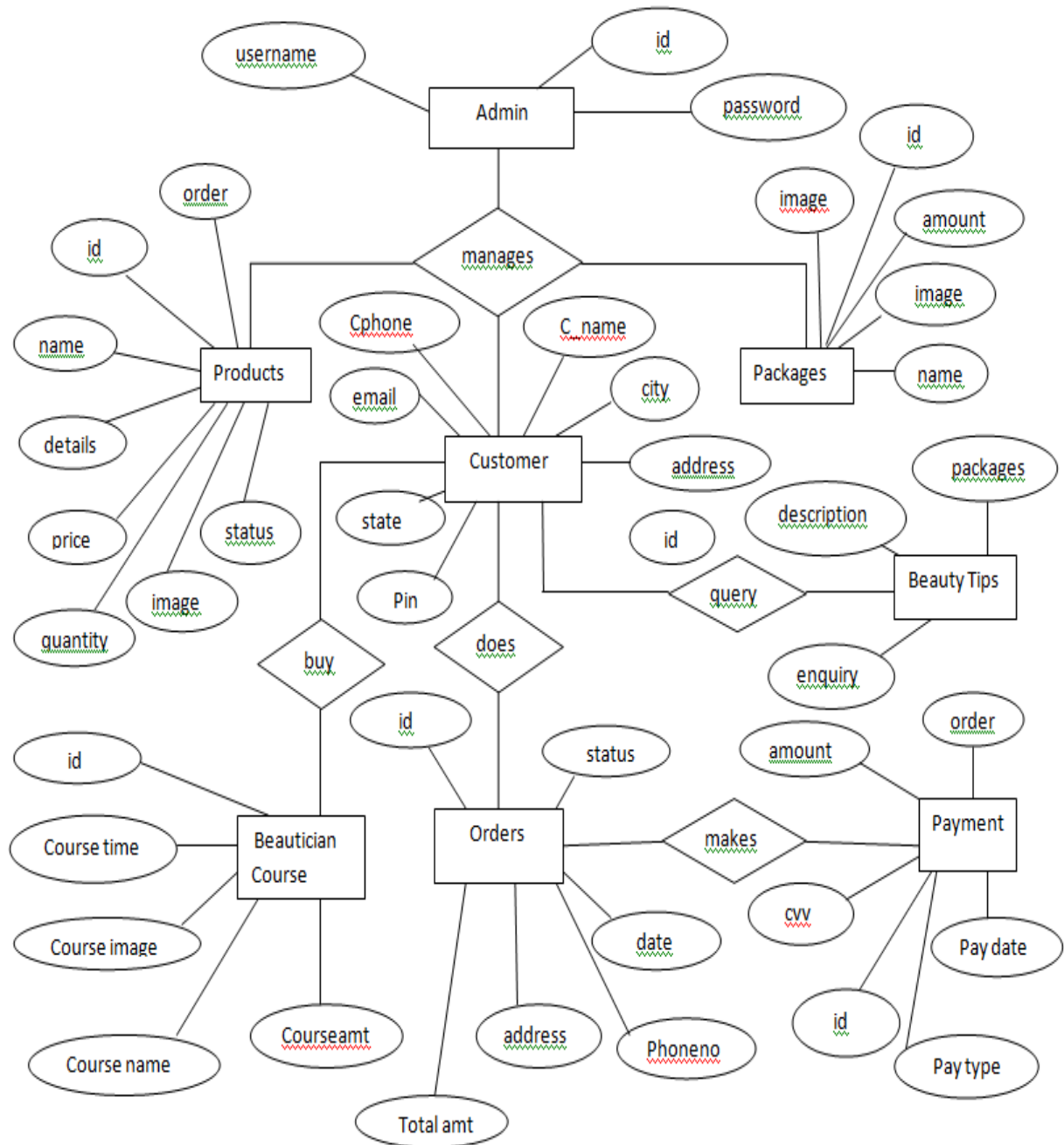


- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.



- **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinarily is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and cordiality specifies the absolute minimum number of relationships.
 - One to One

- One to Many
- Many to One
- Many to Many

E-R DIAGRAM FOR THE BEAUTY HIVE

DATA FLOW DIAGRAM (level 0 and level 1)

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and are transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

PHYSICAL VS LOGICAL DFD

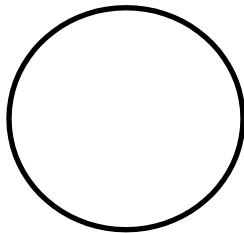
A logical DFD captures the data flows that are necessary for a system to operate. It describes the processes that are undertaken, the data required and produced by each process, and the stores needed to hold the data. On the other hand, a physical DFD shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intends it to be in the future (Required Physical DFD).

Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet. It is quite possible that a Physical DFD will include references to data that are duplicated, or redundant, and that the data stores, if implemented as a set of database tables, would constitute an un-normalized (or de-normalized) relational database. In contrast, a Logical DFD attempts to capture the data flow aspects of a system in a form that has neither redundancy nor duplication.

DATA FLOW SYMBOLS AND THEIR MEANINGS: -

An entity: A source of data or a destination for data.

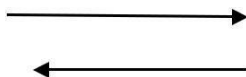
Source/Sink: Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.



Process: Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.



Data Store: Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper.

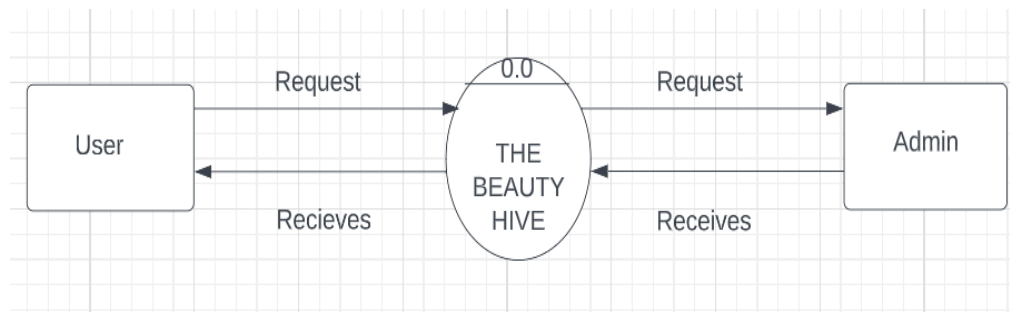


Data Flow: Represented by a unidirectional arrow. Data Flows show how data is moved through the System. Data Flows are labeled with a description of the data that is being passed through it.

A level-0 DFD is the most basic form of DFD. It aims to show how the entire system works at a glance. There is only one process in the system and all the data flows either into or out of this process. Level-0 DFD's demonstrates the interactions between the process and external entities. They do not contain Data Stores.

When drawing Level-0 DFD's, we must first identify the process, all the external entities and all the data flows. We must also state any assumptions we make about the system. It is advised that we draw the process in the middle of the page. We then draw our external entities in the corners and finally connect our entities to our process with the data flows.

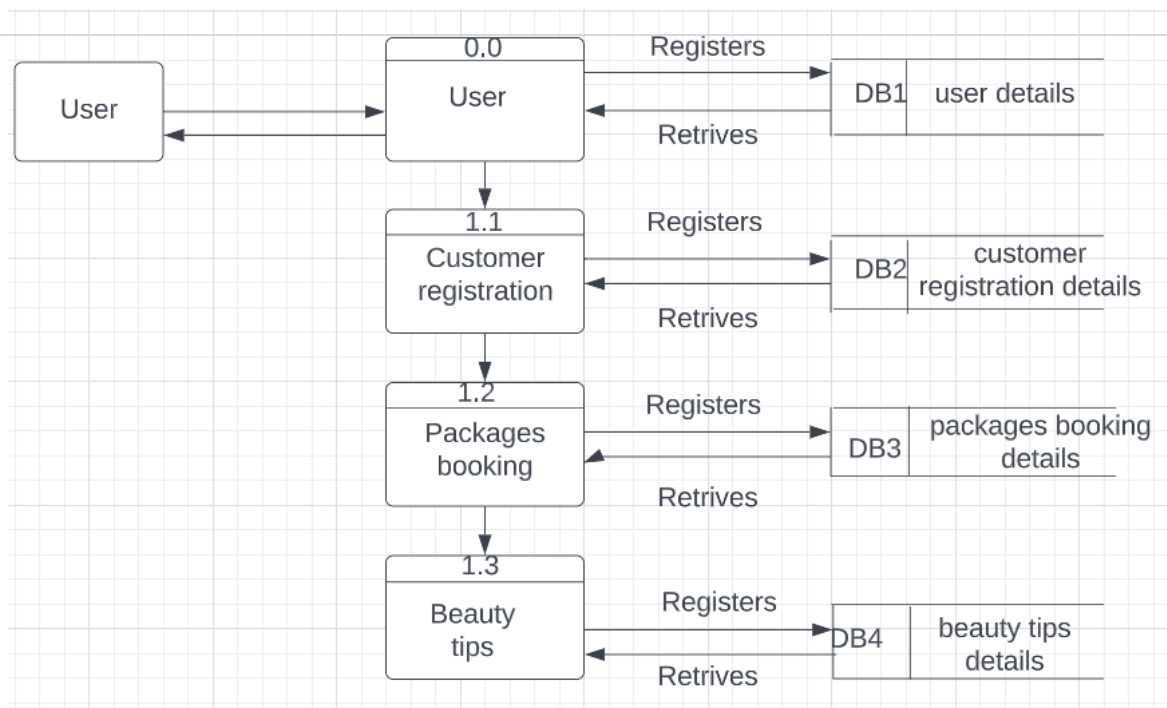
Level 0 DFD/Context Diagram

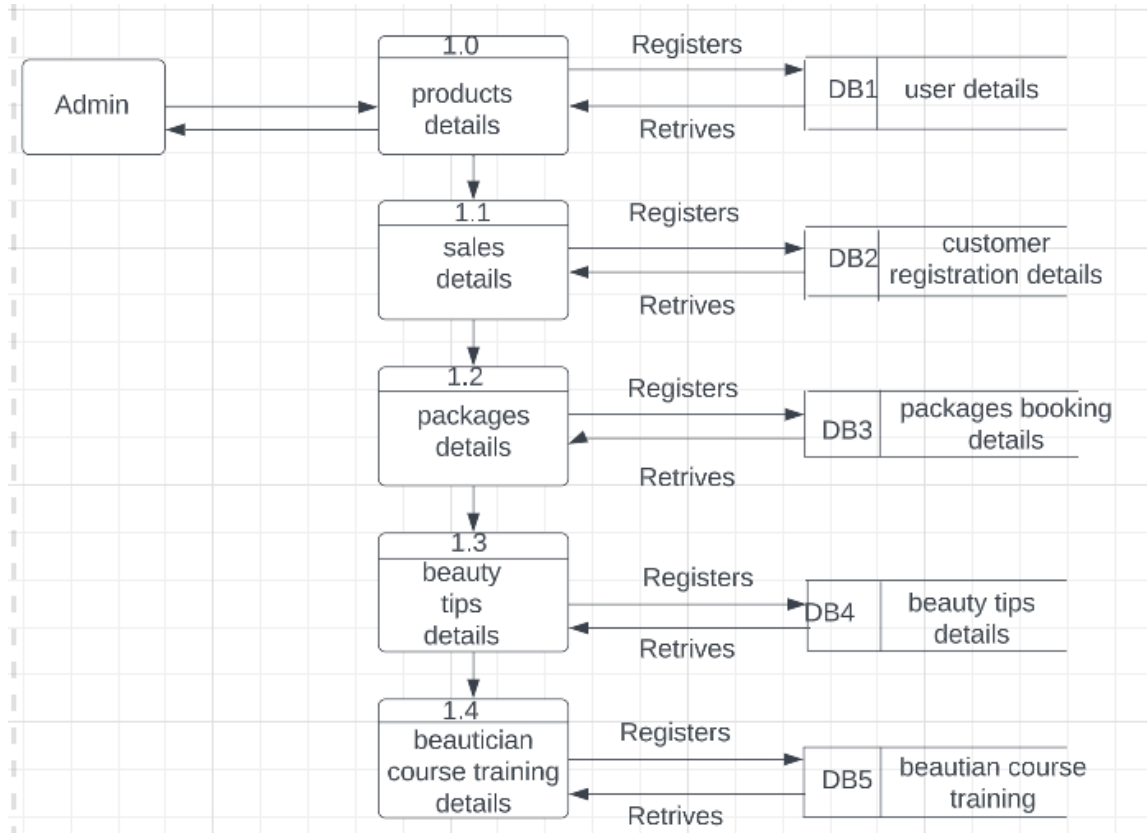


Level 1 DFD:

Level 1 DFD's aim is to give an overview of the full system. They look at the system in more detail. Major processes are broken down into sub-processes. Level 1 DFD's also identifies data stores that are used by the major processes. When constructing a Level 1 DFD we must start by examining the Context Level DFD. We must break up the single process into its sub-processes. We must then pick out the data stores from the text we are given and include them in our DFD. Like the Context Level DFD's, all entities, data stores and processes must be labeled. We must also state any assumptions made from the text.

User:

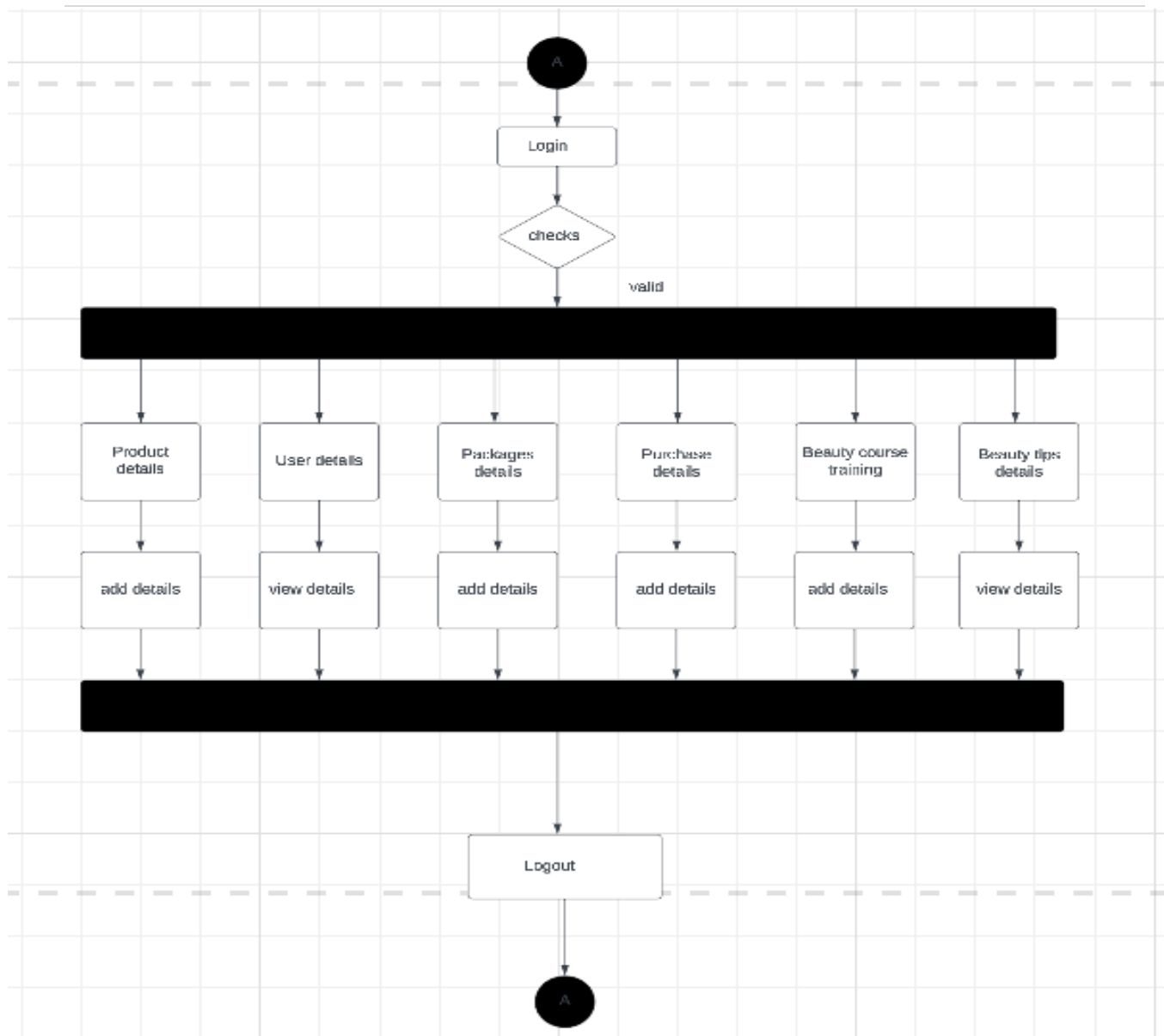


Admin:

ACTIVITY DIAGRAM

An activity diagram visually represents the series of actions or flow of control in a system similar to a flow chart or data flow diagram. Activity diagram are often used in business processing modeling. They can also describe steps in a use case diagram. The activity diagram for Admin module and User module of College Gadget Booking is given below.

ADMIN



USER

GANTT CHART:

A Gantt chart is a type of bar chart, devised by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities.

HISTORICAL DEVELOPMENT:

The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a Harmon gram. Adamiecki did not publish his chart until 1931, however, and only in Polish, which limited both its adoption and recognition of his authorship. The chart is named after Henry Gantt (1861–1919), who designed his chart around the years 1910–1915. One of the first major applications of Gantt charts was by the United States during World War I, at the instigation of General William Crozier in the 1980s, personal computers allowed widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers. With the advent of the Internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative groupware.

GANTT CHART BENEFITS:**Clarity:**

One of the biggest benefits of a Gantt chart is the tool's ability to boil down multiple tasks and timelines into a single document. Stakeholders throughout an organization can easily understand where teams are in a process while grasping the ways in which independent elements come together toward project completion.

Communication:

Teams can use Gantt charts to replace meetings and enhance other status updates. Simply clarifying chart positions offers an easy, visual method to help team members understand task progress.

Motivation:

Some teams or team members become more effective when faced with a form of external motivation. Gantt charts offer teams the ability to focus work at the front of a task timeline, or at the tail end of a chart segment. Both types of team members can find Gantt charts meaningful as they plug their own work habits into the overall project schedule.

Coordination:

For project managers and resource schedulers, the benefits of a Gantt chart include the ability to sequence events and reduce the potential for overburdening team members. Some project managers even use combinations of charts to break down projects into more manageable sets of tasks.

Creativity:

Sometimes, a lack of time or resources forces project managers and teams to find creative solutions. Seeing how individual tasks intertwine on Gantt charts often encourages new partnerships and collaborations that might not have evolved under traditional task assignment systems.

Time Management:

Most managers regard scheduling as one of the major benefits of Gantt charts in a creative environment. Helping teams understand the overall impact of project delays can foster stronger collaboration while encouraging better task organization.

Flexibility:

Whether you use Excel to generate Gantt charts or you load tasks into a more precise chart generator, the ability to issue new charts as your project evolves lets you react to unexpected changes in project scope or timeline. While revising your project schedule too frequently can eliminate some of the other benefits of Gantt charts, offering a realistic view of a project can help team members recover from setbacks or adjust to other changes.

Manageability:

For project managers handling complex assignments, like software publishing or event planning, the benefits of Gantt charts include externalizing assignments. By visualizing all of the pieces of a project puzzle, managers can make more focused, effective decisions about resources and timetables.

Efficiency:

Another one of the benefits of Gantt charts is the ability for teams members to leverage each other's deadlines for maximum efficiency. For instance, while one team member waits on the outcome of three other tasks before starting a crucial piece of the assignment, he or she can perform other project tasks. Visualizing resource usage during projects allows managers to make better use of people, places, and things.

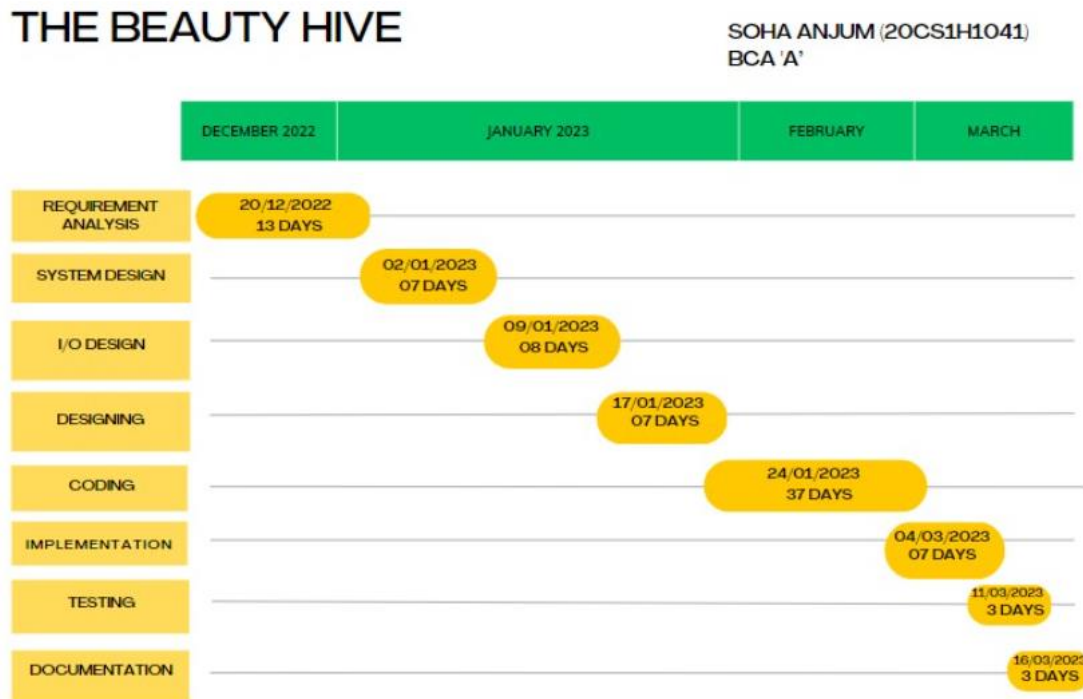
Accountability:

When project teams face major organizational change, documenting effort and outcomes becomes crucial to career success. Using Gantt charts during critical projects allows both project managers and participants to track team progress, highlighting both big wins and major failures during professional review periods; team members who frequently exceed expectations can leverage this documentation into larger raises or bonuses.

Gantt chart Importance:

The project's summary and terminal elements, which combine to form the project's internal structure, are shown on the Gantt chart. Many charts will also depict the precedence rankings and dependencies of various tasks within the project. The charts can illustrate the start and finish project terminal elements in project management. It can also show summary elements and terminal dependencies. The smallest task tracked as part of the project effort is known as a terminal element. Gantt chart represents the tasks in most modern project scheduling packages. However other management applications use simpler communication tools such as message boards, to-do lists and simple scheduling etc., therefore, they do not use Gantt charts as heavily.

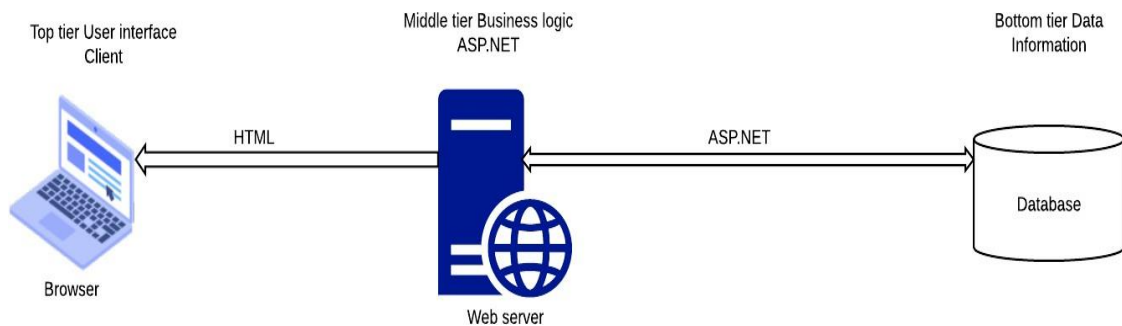
The way to create this chart begins by determining and listing the necessary activities. Next, sketch out how you expect the chart to look. List which items depend on others and what activities take place when. For each activity, list how many man-hours it will require, and who is responsible. Lastly, determine the throughput time.



This technique's primary advantage is its good graphical overview that is easy to understand for nearly all project participants and stakeholders. Its primary disadvantage is its limited applicability for many projects, since projects are often more complex than can be effectively communicated with this chart.

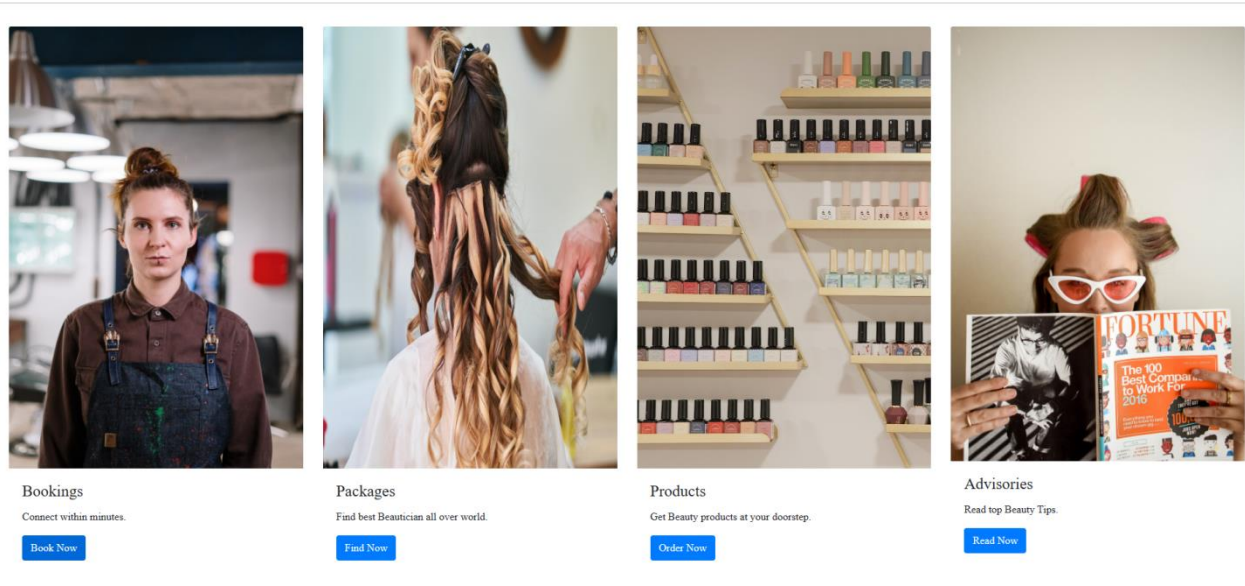
3.5. ARCHITECTURAL DESIGN

An architectural model (in software) is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem.



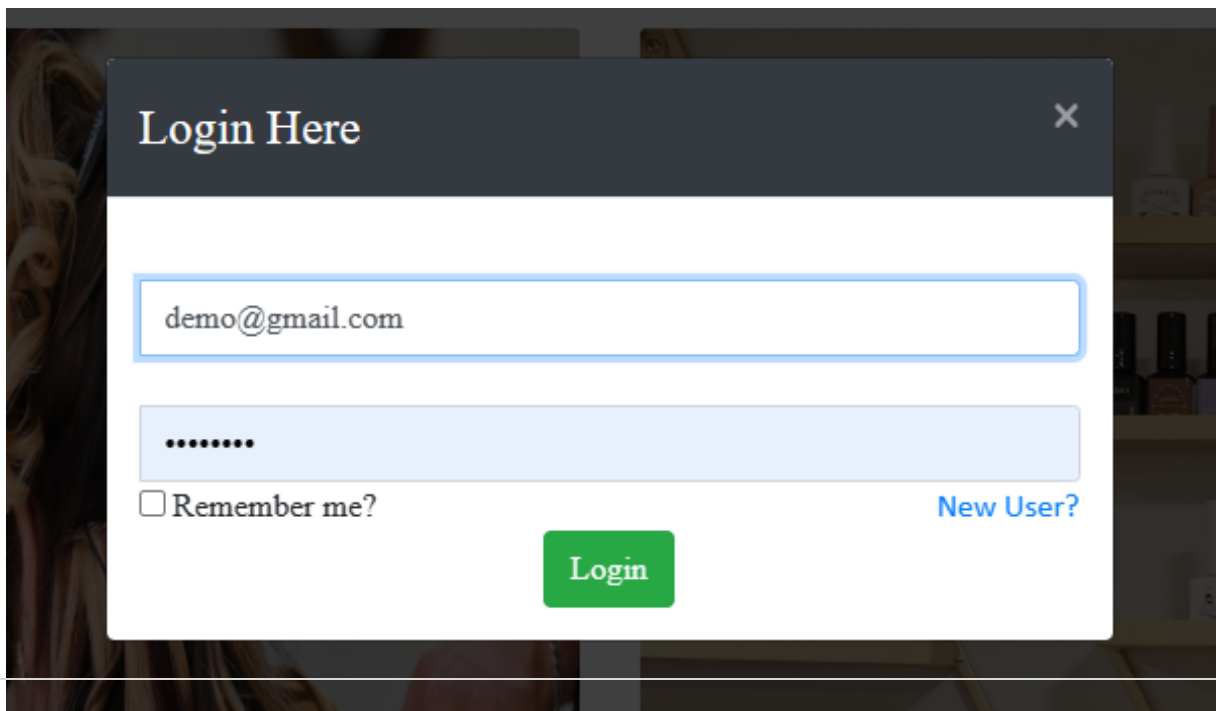
3.6 INPUT DESIGN

Homepage



User Login

Products Details



Admin Login

Welcome, Admin

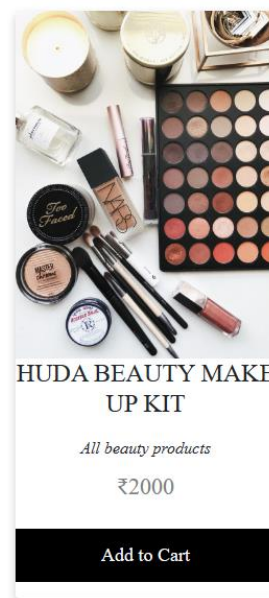
Email: admin@gmail.com

Password:

Update

Products Details

Products




Book Package

Name:

Demo Sample

Appointment Date:

dd-mm-yyyy 

Appointment Time:

--:-- 

Package:

Select Package



Select Type:



Mode of Appointment:

☐ Home ☐ Salon

Mode of Payment:

☐ Card
☐ UPI

Schedule Booking

Beautician Course Training Details:

Makeup Artist

Premium skin care services like Q-Switch, Insta Glow & Insta Clarity Peels, Luxury Facials, and much more included in our bridal pack for a glowing skin.

₹11000

[Book Appointment](#)



Our Packages:

Our Packages

Bridal Make Up

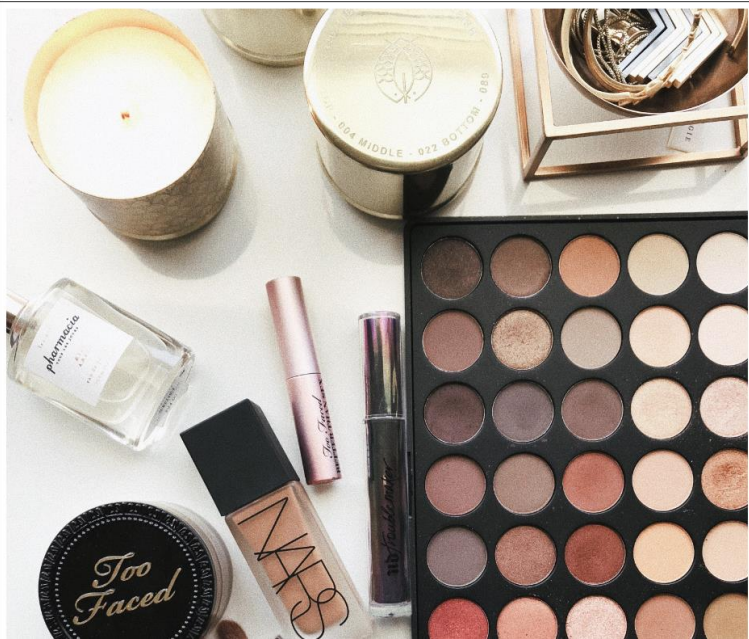
Premium skin care services like Q-Switch, Insta Glow & Insta Clarity Peels, Luxury Facials, and much more included in our bridal pack for a glowing skin.

₹21000

[Book Appointment](#)



Products Details:

| Product ID | Product Image | Product Name | Product Price | Description | Action |
|------------|--|-------------------------|---------------|---------------------|------------------------|
| 30 |  | HUDA BEAUTY MAKE UP KIT | 2000 | All beauty products | Remove |

4.SYSTEM CONFIGURATION

4. SYSTEM CONFIGURATION

Hardware requirements:

| | |
|------------------|---------|
| RAM | 8.00GB |
| Hard disk | 500GB |
| Processor | Pentium |
| Processing speed | 1.80GHz |

Software Requirements:

| | |
|------------------|---------------------------|
| Front end | Visual Studio Code |
| Back end | Microsoft SQL Server 2008 |
| Tools | PHP,SQL SERVER |
| Operating System | Windows 10 |
| Documentation | Microsoft Word 2007 |

5.DETAILS OF THE SOFTWARE

5. DETAILS OF THE SOFTWARE

A development process consists of various phases, each phase ending with a defined output. The phases are performed in an order specified by the process model being followed. The main reason for having a phased process is that it breaks the problem of developing software into successfully performing a set of phase, each handling a different concern of software development.

This ensures that the cost of development is lower than what it would have been if the whole problem were tackled together. A phased development process is central to the software engineering approach for solving the software crisis

Overview of Front end

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE,.NET Framework,

.NET Compact Framework and Microsoft Silver light. Visual Basic 9.0 was released together with the Microsoft .NET Framework 3.5 on 19 November 2007. ASP.NET is a set of Web development tools offered by Microsoft. Programs like Visual Studio .NET and Visual Web Developer allow Web developers to create dynamic websites using a visual interface. ASP.NET also supports Visual Basic .NET, Jscript. .NET and open-source languages like Python and Perl. .NET provide a visual interface for developers to create their applications, which makes .NET a reasonable choice for designing Web-based interfaces as well. SQL Server is a relational database developed and sold by Microsoft. Originally bought from Sybase, Microsoft has released versions 6, 6.5, 7, 2000 and 2005. The Express Edition of SQL Server 2005 is used in the C++ and VB.

Overview of Back-end

Introduction to SQL Server 2008

The Structured Query Language (SQL) comprises one of the fundamental building blocks of modern database architecture. SQL defines the methods used to create and manipulate relational databases on all major platforms.

SQL comes in many flavors. Oracle databases utilize their proprietary PL / SQL. Microsoft SQL Server makes use of Transact – SQL. However, all of these variations are based upon the industry standard ANSI SQL.

SQL commands can be divided into two main sub languages. The Data Definition Language (DDL) contains the commands used to create and destroy databases and database objects. After the database structure is defined with DDL, database administrators and users can

utilize the Data Manipulation Language to insert, retrieve and modify the data contained within it.

SQL Server 2008 is a powerful tool for turning information into opportunity. Industry leading support for XML, enhanced tools for system management and tuning, and exceptional scalability and reliability make SQL Server 2008 the best choice for the agile enterprise.

About the Platform

Windows is a series of Operating Systems developed by Microsoft. Each version of Windows includes a Graphical User Interface, with a desktop that allows users to view files and folders in Windows. For the past two decades, Windows has been the most widely used operating system for personal computers PCs.

Microsoft Windows is designed for both home computing and professional purposes. Past versions of Windows home editions include Windows 3.0 (1990), Windows 3.1 (1992), Windows 95 (1995), Windows 98 (1998), Windows Me (2000), Windows XP (2001), and Windows Vista (2006). The current version, Windows 7, was released in 2009.

The first business-oriented version of Windows, called Windows NT 3.1, was in 1993. This was followed by Windows 3.5, 4.0, and Windows 2000. When Microsoft released Windows XP in 2001, the

Company simply created different editions of the operating system for personal and business purposes. Windows Vista and Windows 7 have followed the same release strategy.

Windows is designed to run on standard x86 hardware, such as Intel and AMD processors. Therefore, it can be installed on multiple brands of hardware, such as Dell, HP, and Sony computers, as well as home-built PCs. Windows 7 also includes several touch screen features, that allow the operating system to run on certain tablets and computers with touch screen displays. Microsoft's mobile operating system, Windows Phone, is designed specifically for smart phones and runs on several brands of phones, including HTC, Nokia, and Samsung.

.NET FRAMEWORK

.NET Framework (pronounced as "dot net") is a software frame work developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named Common Language Runtime(CLR), an application virtual machine that provides services such as security, memory management, and exception handling. As such, computer code written using .NET Framework is called "managed code".

FCL provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their source code with .NET Framework and other libraries. The framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

.NET Framework began as proprietary software, although the firm worked to standardize the software stack almost immediately, even before its first release. Despite the standardization efforts, developers, mainly those in the free and open-source software communities, expressed their unease with the selected terms and the prospects of any free and open-source implementation, especially regarding software patents. Since then, Microsoft has changed .NET development to more closely follow a contemporary model of a community- developed software project, including issuing an update to its patent promising to address the concerns.

6. TESTING

6. TESTING

Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements. To get the most useful perspective on your development project, it is worthwhile devoting some thought to the entire lifecycle including how feedback from users will influence the future of the application. The tools and techniques we've discussed in this book should help your team to be more responsive to changes without extra cost, despite the necessarily wide variety of different development processes. Nevertheless, new tools and process improvements should be adopted gradually, assessing the results after each step.

Testing is part of a lifecycle. The software development lifecycle is one in which you hear of a need, you write some code to fulfill it, and then you check to see whether you have pleased the stakeholders—the users, owners, and other people who have an interest in what the software does. Hopefully they like it, but would also like some additions or changes, so you update or augment your code; and so the cycle continues, or every few years,

SOFTWARE DEVELOPMENT LIFE CYCLE

Testing is a proxy for the customer. You could conceivably do your testing by releasing it into the wild and waiting for the complaints and compliments to come back. Some companies have been accused of having such a strategy as their business model even before it became fashionable. But on the whole, the books are better balanced by trying to make sure that the software will satisfy the customer before we hand it over. This portal “**THE BEAUTY HIVE**” is developed using Incremental Model and Spiral Model.

SOFTWARE TESTING TYPES:

1. FUNCTIONAL TESTING – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

They are:

Black box testing – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

White box testing – This testing is based on knowledge of the internal logic of an application's code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Grey box testing – Also called Grey box analysis, is a strategy for software debugging in which the tester has limited knowledge of the internal details of the program. A Grey box is a device, program or system whose workings are partially understood.

Unit testing – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. May require developing test drive modules or test harnesses.

System testing – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

Acceptance testing -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application.

Alpha testing – In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

2. NON-FUNCTIONAL TESTING

Security testing – Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks.

Usability testing – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point, basically system navigation is checked in this testing.

7.CONCLUSION AND FUTURE ENHANCEMENT

7. CONCLUSION AND FUTURE ENHANCEMENT

Finally in the beauty hive , we have a system where we can a user who is eager to get the services of the beauty at the time and want to do the specific task at the venue. He specifies the materials they want get managed by the organization.

As he confirms after doing the necessary payment it will be forwarded to the management of the beauty hive they get the bookings done to clear and they do the same with the arrangements done. User comes up to the beauty store at a time get the services done.

This project can be further enhanced as follows:

- Minimizing the hardware and software requirement so that it supports maximum user base.
- Modifying the project with better approach, with more graphics.
- The Users can easily communicate with the service providers, in case of any problem or enquiries.
- Security could be heightened.

8.BIBILIOGRAPHY

8. BIBLIOGRAPY

BOOK REFERENCES:

1. The Complete Reference Visual Basic.Net|| 2002 Edition by Jeffer R. Shipario published by TateMcGraw-Hill.
2. Visual Basic .Net Programming||2002 Edition by Peter Aikten published by Dream Tech New Delhi.
3. Programming Microsoft ASP.NET 3.5”edition by Dino Esposito.
4. Visual Basic.net Programming Black Book|| 2005 Edition by Steven Holzer published by Para glyph Press USA.

WEB REFERENCES:

- <https://phpgurukul.com/beauty-parlour-management-system-using-php-and-mysql/>
- <https://sites.google.com/site/ignoubcafinalyearprojects/project-report/beauty-parlour-management-system-project-report>
- <https://www.scribd.com/document/438208917/beauty-parlour-management-system>
- https://www.irjmets.com/uploadedfiles/paper//issue_3_march_2022/19779/final/fin_irjmets1648557623.pdf
- <https://www.studocu.com/in/document/uka-tarsadia-university/business-research/beauty-parlour-management-system-pdf-free/40068412>

9. APPENDICES A-Table Structure

9. APPENDICES A-Table Structure

1.Table Name :Registration

| Attributes | Constraint Name | Data Type |
|----------------|-----------------|--------------|
| User Id | Primary Key | nvarchar(50) |
| User Name | | nvarchar(50) |
| DOB | | Date |
| Address | | nvarchar(50) |
| Contact Number | | nvarchar(50) |
| Email | | nvarchar(50) |
| Password | | nvarchar(50) |

2.Table Name :Packages

| Attributes | Constraint Name | Data Type |
|--------------|-----------------|--------------|
| Username | Primary key | nvarchar(50) |
| Receipt No | | nvarchar(50) |
| Booking Date | | Date |
| Package Name | | nvarchar(50) |
| Mob no | | nvarchar(50) |
| Amount | | nvarchar(50) |
| Payment | | nvarchar(50) |

3.Table Name: Services _Booking

| Attributes | Constraint Name | Data Type |
|-------------------|-----------------|--------------|
| Username | Primary key | nvarchar(50) |
| Registration Date | | Date |
| Booking Type | | nvarchar(50) |
| Function Date | | Date |
| Amount | | nvarchar(50) |
| Payment | | nvarchar(50) |
| Time | | Time |

4.Table Name :Admin

| Attributes | Constraint Name | Data Type |
|------------|-----------------|--------------|
| Username | Primary key | nvarchar(50) |
| Password | | nvarchar(50) |

5.Table Name : Products _Details

| Attributes | Constraint Name | Data Type |
|--------------|-----------------|--------------|
| Product Id | Primary Key | nvarchar(50) |
| Product name | | nvarchar(50) |
| Age | | nvarchar(50) |
| Gender | | nvarchar(50) |
| Address | | nvarchar(50) |
| Date of join | | Date |
| Designation | | nvarchar(50) |
| Mobile | | nvarchar(50) |
| Email | | nvarchar(50) |
| Password | | nvarchar(50) |

6.Beauty Tips:

| Attributes | Constraint Name | Data Type |
|----------------|-----------------|--------------|
| Username | Primary key | nvarchar(50) |
| Ratings | | nvarchar(50) |
| Review Message | | nvarchar(50) |

7.Table Name: Beautician Course Training

| Attributes | Constraint Name | Data Type |
|-------------|-----------------|--------------|
| Course Id | Primary Key | nvarchar(50) |
| Course name | | nvarchar(50) |
| Date | | Date |
| Clock-in | | nvarchar(50) |
| Clock-out | | nvarchar(50) |

8.Table Name: Payment

| Attributes | Constraint Name | Data Type |
|----------------|-----------------|--------------|
| Receipt No | Primary Key | nvarchar(50) |
| Receipt date | | Date |
| Type of income | | nvarchar(50) |
| Amount | | nvarchar(50) |

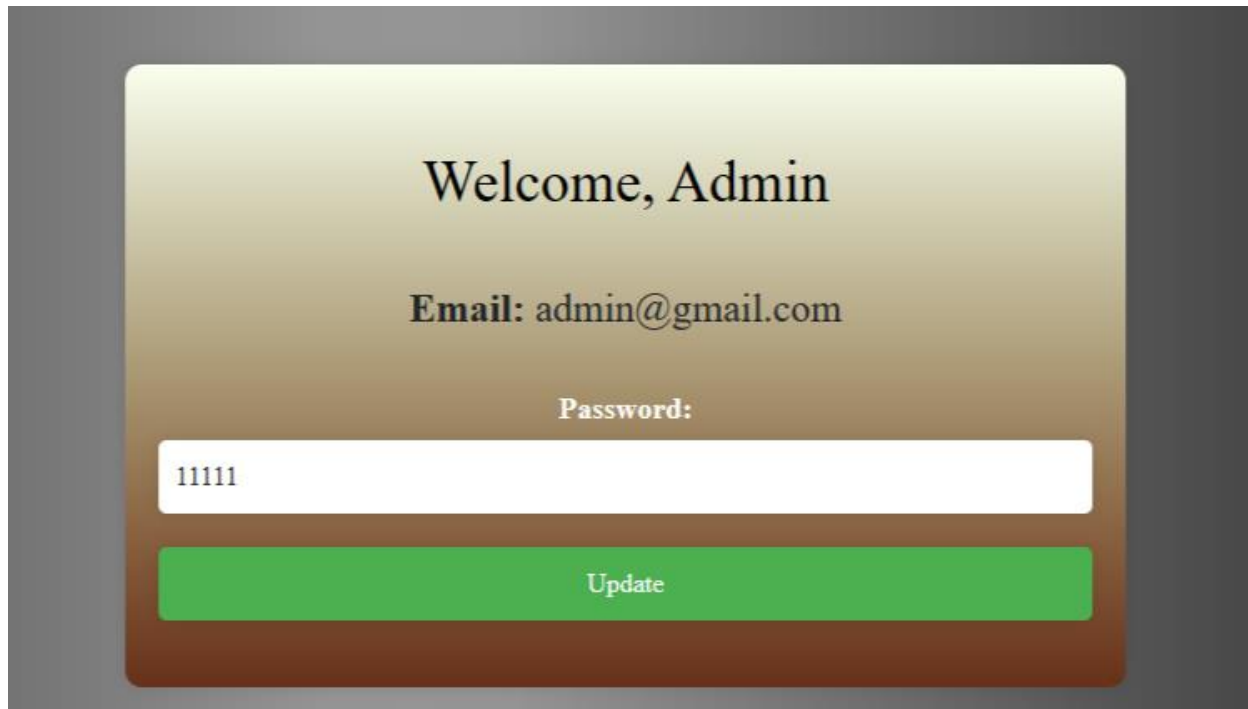
9.Table Name: Purchase_ Details

| Attributes | Constraint Name | Data Type |
|-----------------|-----------------|--------------|
| Purchase id | Primary Key | nvarchar(50) |
| Expense no | | nvarchar(50) |
| Expense date | | Date |
| Type of expense | | nvarchar(50) |
| Amount | | nvarchar(50) |

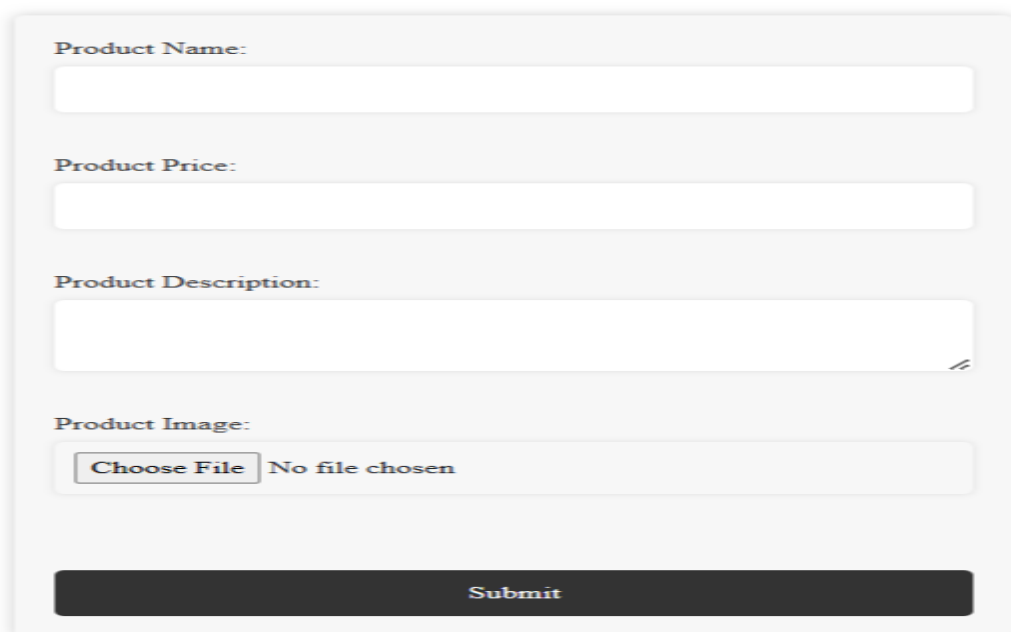
10. Table Name :Sales_ Details

| Attributes | Constraint Name | Data Type |
|--------------|-----------------|--------------|
| Sales id | Primary Key | nvarchar(50) |
| Sales Date | | nvarchar(50) |
| Product | | nvarchar(50) |
| Rate | | nvarchar(50) |
| Quantity | | nvarchar(50) |
| Total amount | | nvarchar(50) |

10.APPENDICES B-SCREENSHOTS

Admin**a.APPENDICES B-SCREENSHOTS**

A screenshot of an admin login interface. It features a central panel with a gradient background transitioning from light yellow at the top to dark brown at the bottom. The panel is set against a dark gray background. The text 'Welcome, Admin' is displayed in a large, black, serif font. Below it, the email 'Email: admin@gmail.com' is shown in a smaller, black, serif font. A 'Password:' label is positioned above a white text input field containing the text '11111'. At the bottom of the panel is a wide, green button with the word 'Update' in white text.

Add Products

A screenshot of an 'Add Products' form. The form is a light gray rounded rectangle with a subtle drop shadow. It contains four labeled input fields: 'Product Name:', 'Product Price:', 'Product Description:', and 'Product Image:'. The 'Product Name', 'Product Price', and 'Product Description' fields are white with rounded ends. The 'Product Image' field is a file upload control showing a 'Choose File' button and the text 'No file chosen'. At the bottom of the form is a wide, dark gray button with the word 'Submit' in white text.

Contact Us

Contact Us

Name

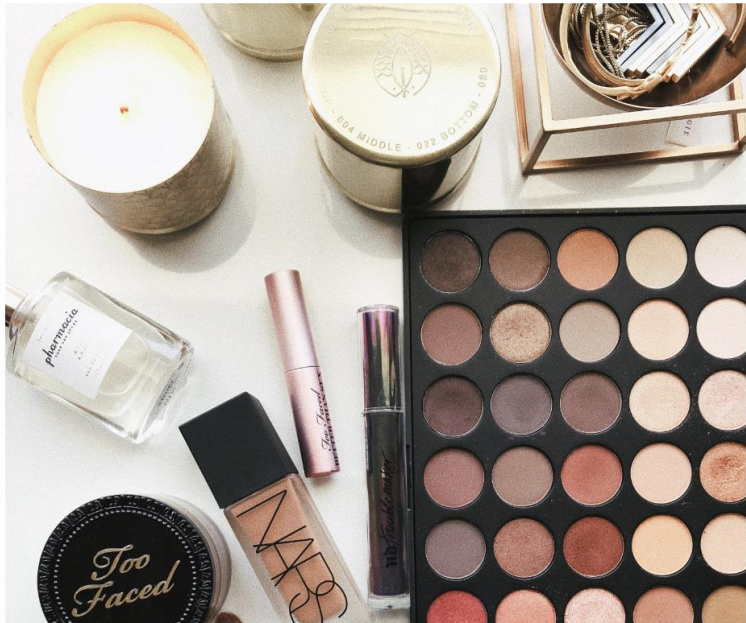
Email ID

Mobile No.

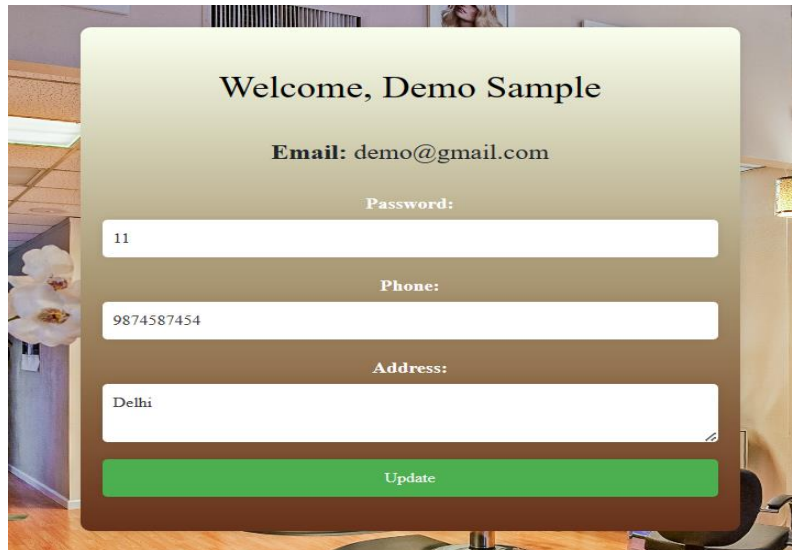
Comments

Submit

Products

| Product ID | Product Image | Product Name | Product Price | Description | Action |
|------------|--|-------------------------|---------------|---------------------|------------------------|
| 30 |  | HUDA BEAUTY MAKE UP KIT | 2000 | All beauty products | Remove |

Profile



Welcome, Demo Sample

Email: demo@gmail.com

Password:

11

Phone:

9874587454

Address:

Delhi

Update

Beautician Course Training

Makeup Artist

Premium skin care services like Q-Switch, Insta Glow & Insta Clarity Peels, Luxury Facials, and much more included in our bridal pack for a glowing skin.

₹11000

[Book Appointment](#)



11.APPENDICES C-SAMPLE TEST CASES

a. APPENDICES C- SAMPLE REPORT AND TEST CASES

| SL. No | Test ID | Form | Test Description | Step to execute | Test Data | Expected Result | Actual Result | Status |
|--------|---------|-------------------|--------------------------|-----------------------------------|------------------|----------------------------|----------------------------|--------|
| 1. | T_01 | admin Form | Check username | Type username | xyz | Invalid input | Invalid input | Pass |
| 2. | T_02 | Admin login | Check username password | Type no password and no user name | | Enter password | Enter password | Pass |
| 3. | T_03 | Admin login | Check user name password | Type user name and no password | demo | Enter password | Enter password | Pass |
| 4. | T_04 | Registration Form | Check ID | Type character | abcd | Enter only numbers | Enter only numbers | Pass |
| 5. | T_05 | Registration form | Check name | Type no name | | Enter the details | Enter the details | Pass |
| 6. | T_06 | Registration Form | Check DOB | Type no DOB | | Enter the details | Enter the details | Pass |
| 7. | T_07 | Registration Form | Check gender | Type no gender | | Enter the gender | Enter the gender | Pass |
| 8. | T_08 | Payment form | Check card number | Type above 16 digit | 1234567890097543 | Enter 16 digit card number | Enter 16 digit card number | Pass |
| 9. | T_09 | Payment form | Check card number | Type below 16 digit | 12345 | Enter 16 digit card number | Enter 16 digit card number | Pass |
| 10. | T_10 | Payment Form | Check CVV number | Type above 3 digit | 8906 | Enter only 3 digits number | Enter only 3 digits number | Pass |

| | | | | | | | | |
|-----|----------|--------------|-----------------------------|--------------------|------------|-------------------------------|-------------------------------|------|
| 11. | T_1 1 | Payment Form | Check CVV number | Type below 3 digit | 89 | Enter only 3 digits number | Enter only 3 digits number | Pass |
| 12. | T_1 2 | Payment Form | Check the expiry date | Type in dd/mm/yyyy | 08/02/2019 | Enter in the format(m m/yyyy) | Enter in the format(m m/yyyy) | Pass |
| 13. | T_1 3 | Product form | Check price from admin side | Type character | abcd | Enter only numbers | Enter only numbers | Pass |
| 14. | T_1 4 | User login | Check username | Type no username | | Please fill the details | Please fill the details | Pass |
| 15. | T_1 5 | User login | Check password | Type no passwords | | Please fill the details | Please fill the details | Pass |

12.APPENDICES D_SOURCE CODE

b. APPENDICES D- SOURCE CODE**Admin Login Page**

```
<!DOCTYPE html>
<html>
<?php
session_start();
include'conn.php';

// Validate the input for $aid
if (!isset($_SESSION['aid'])) {
    echo "Missing required session variable 'aid'.";
    exit();
}

// Retrieve the total number of customers
$totalPatientQuery = "SELECT COUNT(*) as Overall FROM patients";
$totalPatientsResult = mysqli_query($conn, $totalPatientsQuery);
$totalPatientsCount = mysqli_fetch_assoc($totalPatientsResult)['Overall'];

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="home.css" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-awesome.min.css">

    <title>Admin Dashboard</title>
</head>

<body>

    <?php
include'admin_sidebar.php';
?>

    <div class="main-content">
```

```

<div class="row">
  <div class="col-md-6 col-vlg-3 col-sm-6">
    <div class="tiles green m-b-10">
      <div class="tiles-body">
        <div class="tiles-title ">Visitors </div>
        <div class="widget-stats">
          <div class="wrapper transparent">
            <span class="item-title">Overall</span> <span
              class="item-count animate-number semi-bold"
              data-animation-duration="700"><?php echo $totalPatientsCount+10;
?></span>
            </div>
          </div>
        </div>
        <div class="widget-stats ">
          <div class="wrapper last">
            <span class="item-title">Today</span> <span class="item-count animate-
number semi-bold"
              data-animation-duration="700"><?php echo $todayPatientsCount+5;
?></span>
            </div>
          </div>
        </div>
      </div>
    </div>
  <div class="col-md-6 col-vlg-3 col-sm-6">
    <div class="tiles blue m-b-10">
      <a href="view_sales.php" style="color: white;">
        <div class="tiles-body">
          <div class="tiles-title ">Sales</div>
          <div class="widget-stats">
            <div class="wrapper transparent">
              <span class="item-title">Till Now</span> <span
                class="item-count animate-number semi-bold"
                data-animation-duration="700"><?php echo $SalesCount; ?></span>
              </div>
            </div>
          </div>
        </div>
      </a>
    </div>
  </div>
</div>
<?php include 'footer.php' ?>
</body>

```

</html>

Products Page

<!DOCTYPE html>

<html>

<?php

session_start();

include'conn.php';

?>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">

<link rel="stylesheet" type="text/css" href="style.css">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="home.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<title>Admin Dashboard</title>

</head>

<body>

<?php

include'admin_sidebar.php';

?>

<div class="main-content">

<div class="py-5">

<h2 class="text-center">Products Info</h2>

</div>

```
<div class="row">
```

```
<?php
```

```
// Validate the input for $aid
```

```
if (!isset($_SESSION['aid'])) {
```

```
    echo "Missing required session variable 'aid'.";
```

```
    exit();
```

```
}
```

```
$aid = mysqli_real_escape_string($conn, $_SESSION['aid']);
```

```
// Query the doctors table
```

```
$sql = "SELECT * FROM product ORDER BY pro_id ASC";
```

```
$result = mysqli_query($conn, $sql);
```

```
// Check if any doctors were found
```

```
if (mysqli_num_rows($result) > 0) {
```

```
    // Output the doctor data in a table format
```

```
    echo "<table>
```

```
        <tr>
```

```
            <th>Product ID</th>
```

```
            <th>Product Image</th>
```

```
            <th>Product Name</th>
```

```
            <th>Product Price</th>
```

```
            <th>Description</th>
```

```
            <th>Action</th>
```

```
        </tr>";
```

```
while($row = mysqli_fetch_assoc($result)) {
```

```
    echo "<tr>
```

```
        <td>".htmlspecialchars($row["pro_id"])."</td>
```

```
        <td><img class='aboutimg' src='".htmlspecialchars($row["pro_image"])."'></td>
```

```
<td>".htmlspecialchars($row["pro_name"])."</td>
<td>".htmlspecialchars($row["pro_price"])."</td>
<td>".htmlspecialchars($row["pro_desc"])."</td>

<td>

<form action="" method='post'>
    <input type='hidden' name='pro_id' value='".htmlspecialchars($row["pro_id"])."'>
    <input type='submit' name='remove' value='Remove' class='remove-btn'>
</form>
</td>
</tr>";
}
echo "</table>";
} else {
    // Output a message if no doctors were found
    echo "No Result Found.";
}

if (isset($_POST['remove'])) {
    $pro_id = $_POST['pro_id'];

    // Remove the item with the given pro_id from the cart table
    $sql = "DELETE FROM product WHERE pro_id='$pro_id'";

    if ($conn->query($sql) === TRUE) {

    } else {
        echo "Error removing item from cart: " . $conn->error;
```

```
    }  
  }  
  
  // Close the database connection  
  mysqli_close($conn);  
?  
  
  </div>  
  
  </div>  
  
  </body>  
  
</html>
```

Cart Page

```
<?php  
session_start();  
include'conn.php';  
  
$pro_id = $_POST['pro_id'];  
$pid = mysqli_real_escape_string($conn, $_SESSION['pid']);  
  
$sql = "INSERT INTO cart (pro_id,pid) VALUES ('$pro_id','$pid')";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Product added to cart successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}  
  
$conn->close();  
?>
```



```
<!DOCTYPE html>
<html>
<?php
session_start();
include'conn.php';
?>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="style.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="home.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-awesome.min.css">

  <title>Admin Dashboard</title>
  <style>
/* Style the label */
label {
  margin-right: 10px;
  margin-bottom: 0rem;
}

/* Style the form container */
.form-group {
  display: flex;
  align-items: center;
  justify-content: right;
  flex-wrap: wrap;
  margin: 20px 0;
}

/* Style the select dropdown */
select {
  padding: 5px;
  border-radius: 5px;
  border: 1px solid #ccc;
  font-size: 16px;
  margin-right: 10px;
  min-width: 150px;
}

/* Style the button */
button[type="submit"] {
  padding: 5px 10px;
  background-color: #4CAF50;
  color: white;
  border-radius: 5px;
  border: none;
  cursor: pointer;
  font-size: 16px;
}

/* Hover effect for the button */
button[type="submit"]:hover {
```

```

background-color: #3e8e41;
}
</style>
</head>

<body>

<?php
include'admin_sidebar.php';
?>

<div class="main-content">
    <div class="py-5">
        <h2 class="text-center">Sales Report</h2>
    </div>

    <!-- Add a form to select the sort criteria -->
    <form method="GET">
        <div class="form-group">
            <label for="sort" class="fa fa-filter fa-2x" aria-hidden="true"> </label>
            <select name="sort" id="sort">
                <option value="order_id">Order ID</option>
                <option value="pro_name">Product Name</option>
                <option value="patient_name">Patient Name</option>
                <option value="posting_date">Date & Time</option>
                <option value="payment">Payment</option>
            </select>
            <button type="submit">Sort</button>
        </div>
    </form>

```

Appointment Page:

```

<?php

session_start();

include'conn.php';

// Get the form data
$pid = mysqli_real_escape_string($conn, $_SESSION['pid']);
$patient_name = $_SESSION['patient_name'];
$appointment_date = $_POST['appointment_date'];
$appointment_time = $_POST['appointment_time'];
$doc_id = $_POST['doc_id'];
$mode = $_POST['mode'];
$payment = $_POST['payment'];

// Insert the appointment data into the database
$sql = "INSERT INTO appointments (pid,patient_name, appointment_date, appointment_time, doc_id,

```

```

mode, payment)
    VALUES ('$pid', '$patient_name', '$appointment_date', '$appointment_time', '$doc_id',
'$mode', '$payment')");

    if (mysqli_query($conn, $sql)) {
        echo '<script>alert("Congratulations, Appointment Scheduled
Successfully!");window.location.href="home.php";</script>';
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Registration Page:

```

<?php
$patient_name = $_POST["patient_name"];
$patient_email = $_POST["patient_email"];
$patient_password = $_POST["patient_password"];
$mobile = $_POST["mobile"];
$address = $_POST["address"];

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "look";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO patients (patient_name, patient_email, patient_password, mobile, address)
VALUES ('$patient_name', '$patient_email', '$patient_password', '$mobile', '$address')";

if ($conn->query($sql) === TRUE) {
    echo '<script>alert("Congratulations,Registration
completed!");window.location.href="home.php";</script>';
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();

```

Order Page:

```

<!DOCTYPE html>
<html>
<?php
session_start();
include'conn.php';
?>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="style.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="home.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

  <title>Orders</title>
  <style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
  }

  table {
    width: 95%;
    border-collapse: separate;
    margin: 0 auto;
  }

  th,
  td {
    border: 1px solid black;
    padding: 8px;
    text-align: center;
  }

  th {
    background-color: #f2f2f2;
  }
  /* Style the label */
  label {
    margin-right: 10px;
    margin-bottom: 0rem;
  }

```