**CHAPTER-1**

**INTRODUCTION**

**ORGANISATION OVERVIEW**

* 1. **INTRODUCTION**

**Cognizant** is an American multinational corporation that provides IT services, including digital, technology, consulting, and operations services. It is headquartered in Teaneck, New Jersey, United States of America. Cognizant is included in the NASDAQ-100 and the S&P 500 indices. It is also one of the fastest growing Fortune 500 companies. It was founded as an in-house technology unit of DUN & BRADSTREEIN in 1994, and started serving external clients in 1996.

Cognizant had a period of fast growth during the 2000s, becoming a Fortune 500 company in 2011. In 2015, the FORTUNE Magazine named it as the world's fourth most admired IT Services Company. In 2017, Cognizant was named in Fortune’s Future 50 list.

Cognizant provides information technology, information security, consulting, ITO and BPO services. These include business & technology consulting, system integration, application development & maintenance, IT infrastructure services, analytics, business intelligence data , warehousing, customer relationship management, supply chain management, engineering & manufacturing solutions, enterprise resource planning, research and development, outsourcing, and testing solutions.

Cognizant has three key practice areas that span its business — Digital Business, Digital Operations, and Digital Systems & Technology.

The company is split into two new major services, Nelson Media Research, and IMS Health. After some time, it became the public subsidiary of the IMS Health. But in 2003, Cognizant sold all its shares in the subsidiary and the CEO also resigned from his post. The company expanded its work from IT services to Outsourcing and business consulting as well. There was a fast growth in the success of Cognizant. The services provided include application development, business intelligence, supply chain management, CRM, etc.Cognizant also follows a global model for delivery. There are many offshore offices of the companies span over many countries in the world. The main sources of business for the company were the American as well as European clients. The company is a provider of high-end services in all fields. It is one of the top 10 companies that are legitimate in bringing the employee on H-1B into the US.

The company also has plans to increase its US business operations in the coming years. A total of 50 recruiters is continually hiring in the United States for different positions in the US itself. The company has nine delivery center in the United States, Bridgewater, New Jersey; Phoenix, Arizona, Des Moines Iowa, etc.

Cognizant houses around 150,000 employees and has spanned all over the world. There are many development center of the company as well. There are many units which are divided into vertical and horizontal sections. The vertical unit is for Healthcare, Retail, Banking Services, whereas the horizontal unit focuses on mobile computing, testing, and BPO. More than 100,000 employees work in India. The major revenue of the company is from financial companies and then from healthcare industries.

The company also launched the Go Green initiative in the year 2008. This cause completely focused on recycling, waste management, and energy conservation. Cognizant is the 50th traded companies in the United States.



**Figure 1.1 Organization Overview Image**

**1.2 Business Model:**

Like many other IT services firms, Cognizant follows a [global delivery model](https://en.wikipedia.org/wiki/Global_delivery_model) based on offshore software R&D offshore outsourcing. The company has a number of offshore development centers outside the United States and near-shore centers in the U.S., Europe and South America.

In its early years, Cognizant gained business from a number of American and European companies with the help of the Dun & Bradstreet brand. The company's senior executives envisaged the firm as a provider of high-end customer services on-par with the six contemporary major [system integrators](https://en.wikipedia.org/wiki/System_integrator) ([Accenture](https://en.wikipedia.org/wiki/Accenture), [BearingPoint](https://en.wikipedia.org/wiki/BearingPoint), [Capgemini](https://en.wikipedia.org/wiki/Capgemini), [E&Y](https://en.wikipedia.org/wiki/E%26Y), [Deloitte](https://en.wikipedia.org/wiki/Deloitte) and [IBM](https://en.wikipedia.org/wiki/IBM)), but at lower prices.

Cognizant was ranked at 195 in Fortune top 200 list of 2018. Cognizant's excellence and shift to digital marketing, digital platforms interests and expertise in nearly all the leading database, digital marketing and emerging technologies has fetched it an eminent position across the globe.

* 1. **Objective**

Organization and employees should develop and progress simultaneously for the attainment of mutual goals. So every modern management has to develop the organization through human resource development. Employee upskilling is one of the most important sub-systems of human resource management team.

To facilitate this objective, the need for an application that takes the learning to the employees, in a form that can be consumed by them, easily and effectively, becomes imperative.

The proposed solution will be a web application used by the employees of an Organization to search for eLearning courses that matches their skill/aspirations, launch the courses from any web platforms, and conduct knowledge checks of their learnings.

The application will also assist the Admin and Employees to generate reports of their completed courses, to enable them plan for the next course of actions.

**1.4 Motivation**

This project was chosen with idea of automation of Training Management so that the Users can use this portal easily and efficiently and can easily get notified by any updates from the admin. The proposed system should provide various features for user benefits so that user can easily learn the skill and give their exams.

**1.5 Definition and overview**

The ‘Training management System’ is a web based application.

‘Digi Learn’ is the name given to the web portal.

Features of this web-application are as follows:

* Register to learn

User can register themselves for any skill lessons going to be held in company.

* Get result details

User can get result details of each skills.

* Navigate through skill lesson

User can easy navigate through each lessons in the skill.

* User Management

It helps training coordinator to manage the training details, Trainee Information easily.

**CHAPTER-2**

**SYSTEM DEVELOPMENT LIFE CYCLE MODEL**

**2.1 Introduction**

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.



Fig.2.1.1. SDLC cycle

**2.2 Feasibility Study**

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources. The objective for this phase is not to solve the problem but to acquire a sense of scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. We started the project by listing the possible queries that the user might want to be satisfied. And on these lines we guided the project further.

The three main points, kept in mind at the time of project are:

* Resource Availability (To build it with the given technology and resources)
* Affordable (Given the time and cost constraints of the organization)
* Acceptable (For use by the eventual users of the system

The three major areas to be considered while determining the feasibility of a project are:

* Economic Feasibility
* Technical Feasibility
* Behavioral Feasibility

**2.2.1 Economic Feasibility**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems.

Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies, there is nominal expenditure and economic feasibility for certain.

**2.2.2 Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Training management System’. The current system developed is technically feasible. It is a web based user interface. Thus it provides an easy access to the users. The databases purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

**2.2.3 Behavioral Feasibility**

Proposed projects are beneficial only if they can be turned out into information system that will meet the organizations operating requirements. Behavioral feasibility aspects of the project are to be taken as an important part of the project implementation. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Behavioral feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The behavioral feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes. To ensure success, desired behavioral outcomes must be imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired behavioral behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, behavioral feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

**CHAPTER-3**

**AGILE METHODOLOGY**

**3.1 What is Agile?**

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many [software development methods](https://en.wikipedia.org/wiki/Software_development_methodologies).

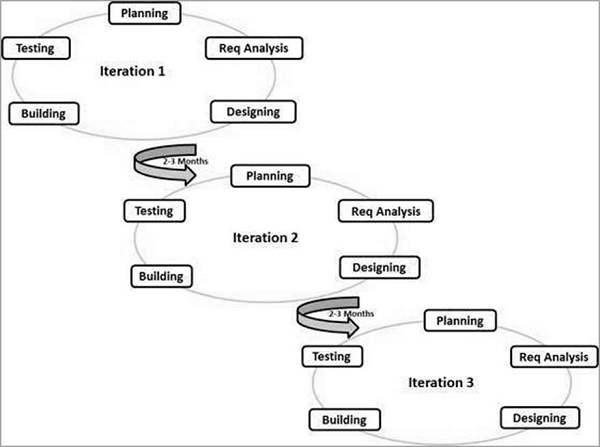


Fig.3.1.1. graphical illustration of the Agile Model

### **3.2 What is Scrum?**

Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.

* A “process framework” is a particular set of practices that must be followed in order for a process to be consistent with the framework. (For example, the Scrum process framework requires the use of development cycles called Sprints, the XP framework requires pair programming, and so forth.)
* “Lightweight” means that the overhead of the process is kept as small as possible, to maximize the amount of productive time available for getting useful work done.

A [Scrum process](https://www.cprime.com/2015/03/5-tips-to-manage-scrum-processes-in-the-real-world/) is distinguished from other agile processes by specific concepts and practices, divided into the three categories of Roles, Artifacts, and Time Boxes. These and other terms used in Scrum are defined below. Scrum is most often used to manage complex software and product development, using iterative and incremental practices. Scrum significantly increases productivity and reduces time to benefits relative to classic “[waterfall](https://www.cprime.com/2011/01/integrating-waterfall-and-agile-development-hybrid-model/)” processes. Scrum processes enable organizations to adjust smoothly to rapidly-changing requirements, and produce a product that meets evolving business goals. An agile Scrum process benefits the organization by helping it to-

* Increase the quality of the deliverables
* Cope better with change (and expect the changes)
* Provide better estimates while spending less time creating them
* Be more in control of the project schedule and state

**3.2.1 Scrum Roles**

Scrum has three roles: Product Owner, Scrum Master, and Team.

* **Product Owner:** The Product Owner should be a person with vision, authority, and availability. The Product Owner is responsible for continuously communicating the vision and priorities to the development team.
  + - It’s sometimes hard for Product Owners to strike the right balance of involvement. Because Scrum values self-organization among teams, a Product Owner must fight the urge to micro-manage. At the same time, Product Owners must be available to answer questions from the team.
* **Scrum Master:** The Scrum Master acts as a facilitator for the Product Owner and the team. The Scrum Master does not manage the team. The Scrum Master works to remove any impediments that are obstructing the team from achieving its sprint goals. This helps the team remain creative and productive while making sure its successes are visible to the Product Owner. The Scrum Master also works to advise the Product Owner about how to maximize ROI for the team.
* **Team:** According to Scrum’s founder, “the team is utterly self-managing.” The development team is responsible for self-organizing to complete work. A Scrum development team contains about seven fully dedicated members (officially 3-9), ideally in one team room protected from outside distractions. For software projects, a typical team includes a mix of software engineers, architects, programmers, analysts, QA experts, testers, and UI designers. Each sprint, the team is responsible for determining how it will accomplish the work to be completed. The team has autonomy and responsibility to meet the goals of the sprint.

### **3.3 Iterative vs. waterfall**

One of the differences between agile software development methods and waterfall is the approach to quality and testing. In the waterfall model there is always a separate *testing phase* after a *build phase*; however, in agile software development testing is completed in the same iteration as programming.

Because testing is done in every iteration—which develops a small piece of the software—users can frequently use those new pieces of software and validate the value. After the users know the real value of the updated piece of software, they can make better decisions about the software's future. Having a value retrospective and software re-planning session in each iteration—Scrum typically has iterations of just two weeks—helps the team continuously adapt its plans so as to maximize the value it delivers.

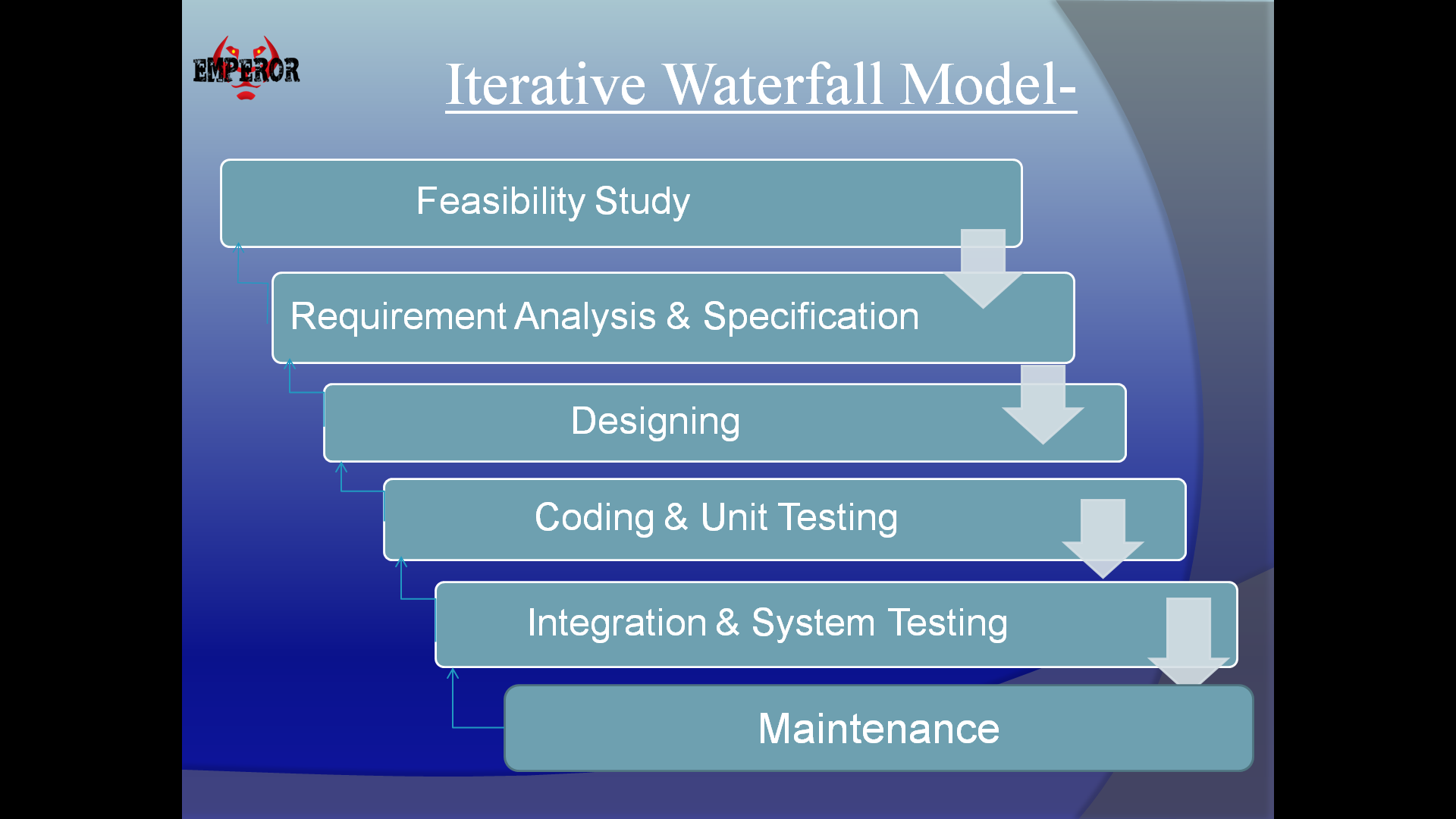


Fig.3.3.1. Phases of Iterative Waterfall Model

This iterative approach supports a *product* rather than a *project* mindset. This provides greater flexibility throughout the development process; whereas on projects the requirements are defined and locked down from the very beginning, making it difficult to change them later. Iterative product development allows the software to evolve in response to changes in business environment or market requirements. Because of the short iteration style of agile software development, it also has strong connections with the lean startup concept.

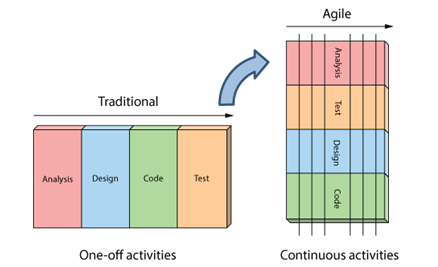
`

Fig.3.3.2.Iterative vs. waterfall

## 3.4 Agile Model - Pros and Cons

Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the agile model.

The advantages of the Agile Model are as follows −

* It is a very realistic approach of software development.
* Promotes teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Suitable for fixed or changing requirements
* Delivers early partial working solutions.
* Good model for environments that change steadily.
* Minimal rules, documentation easily employed.
* Enables concurrent development and delivery within an overall planned context.
* Little or no planning required.
* Easy to manage.
* Gives flexibility to developers.

The disadvantages of the Agile Model are as follows −

* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* An overall plan, an agile leader and agile PM practice is a must without which it will not work.
* Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
* There is a very high individual dependency, since there is minimum documentation generated.
* Transfer of technology to new team members may be quite challenging due to lack of documentation.

**3.5Burn-down Chart**

The burn-down is a chart that shows how quickly you and your team are burning through your customer’s user stories. It shows the total effort against the amount of work we deliver each iteration.

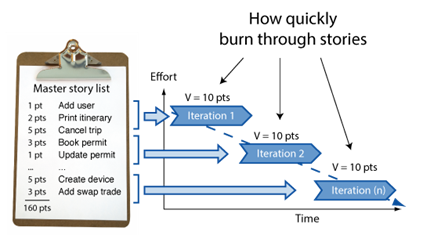


Fig.3.5.1. Working of Iterations

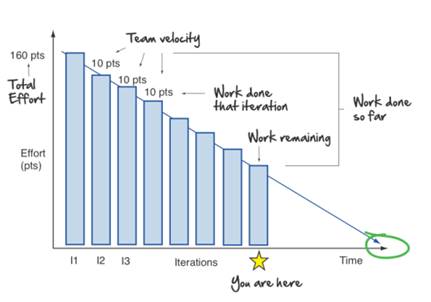


Fig.3.5.2.Ideal Burn Down Graph

We can see the total effort on the left, our team velocity on the right. But look what else this simple graphs gives us.

* Work done each iteration
* Work remaining
* Work done so far
* When we can expect to be done

All this from one graph!

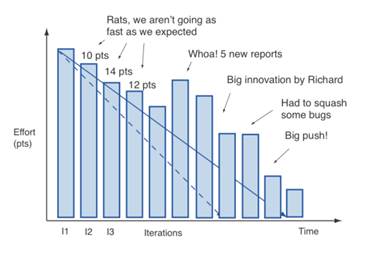
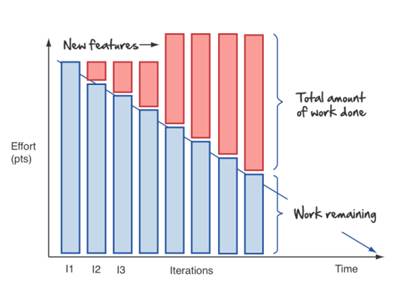
Now what you see above is pretty ideal. A more realistic burn-down looks something like this:

Fig.3.5.3. Realistic Burn Down Graph

It’s never a straight line. The team never moves at exactly one fixed velocity. And we discover things along the way (notice how it shows us scope creep in the form of those 5 new reports).

And of course like all things in Agile, you are free to make things your own. One tweak I like making to the burn-down is displaying total work done each iteration also. This lets me look at the chart, and immediately get a sense of whether we are a quarter, a third, or ½ way done the project.

  
Fig.3.5.4. Explanation

# **3.6 User Stories**

User stories are features our customers might one day like to see in their software.

User stories are like agile requirements except that they’re not. For one there’s no guarantee all these features are going to make it into the final version of the software. Secondly, Agilest know



Fig.3.6.1. Iteration level

their customers are going to change their mind - and that’s OK.

Typically no more than a couple days work, they form the basis of our agile plans.

User stories form the basis of the agile plan. They are sized and prioritized like any other wish list. You simply start at the top and work your way down. Nothing big or complex. Just a prioritized to do list and a desire to get things done.

We get them by sitting down with our customers and asking lots of questions.

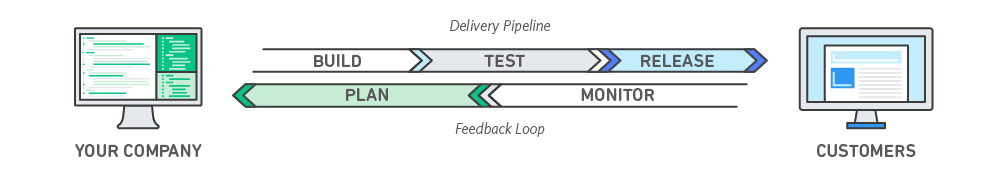
Big rooms with lots of white space to draw are great for gathering user stories. In these story gathering workshops we draw lots of pictures (flowcharts, screens, storyboards, and mockups, anything that helps) and break the functionality down into simple easy to understand words and phrases our customers understand. User stories!

**Chapter 4**

**DevOps**

**Introduction**

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

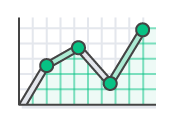


**4.1 How DevOps Works?**

Under a DevOps model, development and operations teams are no longer “siloed.” Sometimes, these two teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function.

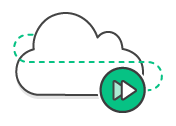
In some DevOps models, quality assurance and security teams may also become more tightly integrated with development and operations and throughout the application lifecycle. When security is the focus of everyone on a DevOps team, this is sometimes referred to as DevSecOps.

These teams use practices to automate processes that historically have been manual and slow. They use a technology stack and tooling which help them operate and evolve applications quickly and reliably. These tools also help engineers independently accomplish tasks (for example, deploying code or provisioning infrastructure) that normally would have required help from other teams, and this further increases a team’s velocity.



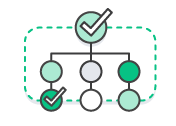
### Speed

Move at high velocity so you can innovate for customers faster, adapt to changing markets better, and grow more efficient at driving business results. The DevOps model enables your developers and operations teams to achieve these results. For example, [microservices](https://aws.amazon.com/devops/what-is-devops/" \l "microservices) and [continuous delivery](https://aws.amazon.com/devops/continuous-delivery/) let teams take ownership of services and then release updates to them quicker.



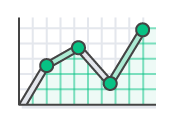
### Rapid Delivery

Increase the frequency and pace of releases so you can innovate and improve your product faster. The quicker you can release new features and fix bugs, the faster you can respond to your customers’ needs and build competitive advantage. [Continuous integration](https://aws.amazon.com/devops/continuous-integration/) and [continuous delivery](https://aws.amazon.com/devops/continuous-delivery/) are practices that automate the software release process, from build to deploy.



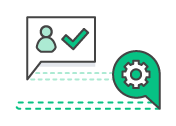
### Reliability

Ensure the quality of application updates and infrastructure changes so you can reliably deliver at a more rapid pace while maintaining a positive experience for end users. Use practices like [continuous integration](https://aws.amazon.com/devops/continuous-integration/) and [continuous delivery](https://aws.amazon.com/devops/continuous-delivery/) to test that each change is functional and safe. [Monitoring and logging](https://aws.amazon.com/devops/what-is-devops/#monitoring) practices help you stay informed of performance in real-time.



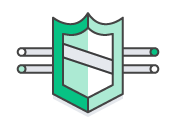
### Scale

Operate and manage your infrastructure and development processes at scale. Automation and consistency help you manage complex or changing systems efficiently and with reduced risk. For example, [infrastructure as code](https://aws.amazon.com/devops/what-is-devops/#iac) helps you manage your development, testing, and production environments in a repeatable and more efficient manner.



Improved Collaboration

Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability. Developers and operations teams [collaborate](https://aws.amazon.com/devops/what-is-devops/#communication) closely, share many responsibilities, and combine their workflows. This reduces inefficiencies and saves time (e.g. reduced handover periods between developers and operations, writing code that takes into account the environment in which it is run).



### Security

Move quickly while retaining control and preserving compliance. You can adopt a DevOps model without sacrificing security by using automated compliance policies, fine-grained controls, and configuration management techniques. For example, using infrastructure as code and [policy as code](https://aws.amazon.com/devops/what-is-devops/#policyascode), you can define and then track compliance at scale.

## 

## 4.2 Why DevOps Matters?

Software and the Internet have transformed the world and its industries, from shopping to entertainment to banking. Software no longer merely supports a business; rather it becomes an integral component of every part of a business. Companies interact with their customers through software delivered as online services or applications and on all sorts of devices. They also use software to increase operational efficiencies by transforming every part of the value chain, such as logistics, communications, and operations. In a similar way that physical goods companies transformed how they design, build, and deliver products using industrial automation throughout the 20th century, companies in today’s world must transform how they build and deliver software.

## 

## 4.3 How to Adopt a DevOps Model?

## A) DevOps Cultural Philosophy

Transitioning to DevOps requires a change in culture and mindset. At its simplest, DevOps is about removing the barriers between two traditionally siloed teams, development and operations. In some organizations, there may not even be separate development and operations teams; engineers may do both. With DevOps, the two teams work together to optimize both the productivity of developers and the reliability of operations. They strive to communicate frequently, increase efficiencies, and improve the quality of services they provide to customers. They take full ownership for their services, often beyond where their stated roles or titles have traditionally been scoped by thinking about the end customer’s needs and how they can contribute to solving those needs. Quality assurance and security teams may also become tightly integrated with these teams. Organizations using a DevOps model, regardless of their organizational structure, have teams that view the entire development and infrastructure lifecycle as part of their responsibilities.

## 

## B) DevOps Practices

There are a few key practices that help organizations innovate faster through automating and streamlining the software development and infrastructure management processes. Most of these practices are accomplished with proper tooling.

One fundamental practice is to perform very frequent but small updates. This is how organizations innovate faster for their customers. These updates are usually more incremental in nature than the occasional updates performed under traditional release practices. Frequent but small updates make each deployment less risky. They help teams address bugs faster because teams can identify the last deployment that caused the error. Although the cadence and size of updates will vary, organizations using a DevOps model deploy updates much more often than organizations using traditional software development practices.

Organizations might also use a micro services architecture to make their applications more flexible and enable quicker innovation. The micro services architecture decouples large, complex systems into simple, independent projects. Applications are broken into many individual components (services) with each service scoped to a single purpose or function and operated independently of its peer services and the application as a whole. This architecture reduces the coordination overhead of updating applications, and when each service is paired with small, agile teams who take ownership of each service, organizations can move more quickly.

However, the combination of micro services and increased release frequency leads to significantly more deployments which can present operational challenges. Thus, DevOps practices like continuous integration and continuous delivery solve these issues and let organizations deliver rapidly in a safe and reliable manner. Infrastructure automation practices, like infrastructure as code and configuration management, help to keep computing resources elastic and responsive to frequent changes. In addition, the use of monitoring and logging helps engineers track the performance of applications and infrastructure so they can react quickly to problems.

Together, these practices help organizations deliver faster, more reliable updates to their customers. Here is an overview of important DevOps practices.

The following are DevOps best practices:

* [Continuous Integration](https://aws.amazon.com/devops/what-is-devops/#integration)
* [Continuous Delivery](https://aws.amazon.com/devops/what-is-devops/#cd)
* [Micro services](https://aws.amazon.com/devops/what-is-devops/#microservices)
* [Infrastructure as Code](https://aws.amazon.com/devops/what-is-devops/#iac)
* [Monitoring and Logging](https://aws.amazon.com/devops/what-is-devops/#monitoring)
* [Communication and Collaboration](https://aws.amazon.com/devops/what-is-devops/#communication)

**CHAPTER-5**

**DETAILS OF TECHNOLOGIES USED**

**5.1 C#**

C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg. This tutorial will teach you basic C# programming and will also take you through various advanced concepts related to C# programming language.

C# was developed by Anders Hejlsberg and his team during the development of .Net Framework.

C# is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages on different computer platforms and architectures.

The following reasons make C# a widely used professional language −

* It is a modern, general-purpose programming language
* It is object oriented.
* It is component oriented.
* It is easy to learn.
* It is a structured language.
* It produces efficient programs.
* It can be compiled on a variety of computer platforms.
* It is a part of .Net Framework.

**5.2 Dot Net(framework)**

**.NET Framework** is a [software framework](https://en.m.wikipedia.org/wiki/Software_framework) developed by [Microsoft](https://en.m.wikipedia.org/wiki/Microsoft) that runs primarily on [Microsoft Windows](https://en.m.wikipedia.org/wiki/Microsoft_Windows). It includes a large [class library](https://en.m.wikipedia.org/wiki/Class_library) named as [Framework Class Library](https://en.m.wikipedia.org/wiki/Framework_Class_Library) (FCL) and provides [language interoperability](https://en.m.wikipedia.org/wiki/Language_interoperability) (each language can use code written in other languages) across several [programming languages](https://en.m.wikipedia.org/wiki/Programming_language). Programs written for .NET Framework execute in a [software](https://en.m.wikipedia.org/wiki/Software) environment (in contrast to a [hardware](https://en.m.wikipedia.org/wiki/Computer_hardware) environment) named the [Common Language Runtime](https://en.m.wikipedia.org/wiki/Common_Language_Runtime) (CLR). The CLR is an [application virtual machine](https://en.m.wikipedia.org/wiki/Process_virtual_machine) that provides services such as security, [memory management](https://en.m.wikipedia.org/wiki/Memory_management), and [exception handling](https://en.m.wikipedia.org/wiki/Exception_handling). As such, computer code written using .NET Framework is called "[managed code](https://en.m.wikipedia.org/wiki/Managed_code)". FCL and CLR together constitute the .NET Framework.

FCL provides [user interface](https://en.m.wikipedia.org/wiki/User_interface), [data access](https://en.m.wikipedia.org/wiki/Data_access), [database connectivity](https://en.m.wikipedia.org/wiki/Database_connection), [cryptography](https://en.m.wikipedia.org/wiki/Cryptography), [web application](https://en.m.wikipedia.org/wiki/Web_application) development, numeric [algorithms](https://en.m.wikipedia.org/wiki/Algorithm), and [network communications](https://en.m.wikipedia.org/wiki/Computer_networking). Programmers produce software by combining their [source code](https://en.m.wikipedia.org/wiki/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](https://en.m.wikipedia.org/wiki/Integrated_development_environment) largely for .NET software called [Visual Studio](https://en.m.wikipedia.org/wiki/Microsoft_Visual_Studio).

**5.3 ASP.NET**

ASP.NET is an [open-source](https://en.m.wikipedia.org/wiki/Open-source_software) [server-side](https://en.m.wikipedia.org/wiki/Server-side_scripting) [web application framework](https://en.m.wikipedia.org/wiki/Web_application_framework) designed for [web development](https://en.m.wikipedia.org/wiki/Web_development) to produce [dynamic web pages](https://en.m.wikipedia.org/wiki/Dynamic_web_page). It was developed by [Microsoft](https://en.m.wikipedia.org/wiki/Microsoft) to allow [programmers](https://en.m.wikipedia.org/wiki/Programmer) to build dynamic [web sites](https://en.m.wikipedia.org/wiki/Web_site), [web applications](https://en.m.wikipedia.org/wiki/Web_application) and [web services](https://en.m.wikipedia.org/wiki/Web_service).

It was first released in January 2002 with version 1.0 of the [.NET Framework](https://en.m.wikipedia.org/wiki/.NET_Framework), and is the successor to Microsoft's [Active Server Pages](https://en.m.wikipedia.org/wiki/Active_Server_Pages) (ASP) technology. ASP.NET is built on the [Common Language Runtime](https://en.m.wikipedia.org/wiki/Common_Language_Runtime) (CLR), allowing programmers to write ASP.NET code using any supported [.NET language](https://en.m.wikipedia.org/wiki/List_of_CLI_languages). The ASP.NET [SOAP](https://en.m.wikipedia.org/wiki/SOAP) extension framework allows ASP.NET components to process SOAP messages.

ASP.NET's successor is [ASP.NET Core](https://en.m.wikipedia.org/wiki/ASP.NET_Core). It is a re-implementation of ASP.NET as a modular [web framework](https://en.m.wikipedia.org/wiki/Web_framework), together with other frameworks like [Entity Framework](https://en.m.wikipedia.org/wiki/Entity_Framework). The new framework uses the new open-source [.NET Compiler Platform](https://en.m.wikipedia.org/wiki/.NET_Compiler_Platform) (codename "Roslyn") and is [cross platform](https://en.m.wikipedia.org/wiki/Cross_platform). [ASP.NET MVC](https://en.m.wikipedia.org/wiki/ASP.NET_MVC), ASP.NET Web API, and ASP.NET Web Pages (a platform using only [Razor](https://en.m.wikipedia.org/wiki/ASP.NET_Razor) pages) have merged into a unified MVC 6.

**5.4 Java Scripts**

JavaScript often abbreviated as "JS", is a high-level, dynamic, un-typed, and interpreted run-time language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production; the majority of websites employ it, and all modern Web browsers support it without the need for plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon objects made available by the host environment in which it is embedded.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two are distinct languages and differ greatly in their design. JavaScript was influenced by programming languages such as self and Scheme.

JavaScript is also used in environments that are not Web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side Web applications. On the client side, developers have traditionally implemented JavaScript as an interpreted language, but more recent browsers perform just-in-time compilation. Programmers also use JavaScript in video-game development and in desktop and mobile applications.

**5.5 Bootstrap**

**Bootstrap** is a  [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software" \o "Free and open-source software) front-end [web framework](https://en.wikipedia.org/wiki/Web_framework) for designing [websites](https://en.wikipedia.org/wiki/Website) and [web applications](https://en.wikipedia.org/wiki/Web_application). It contains [HTML](https://en.wikipedia.org/wiki/HTML)- and [CSS](https://en.wikipedia.org/wiki/CSS)-based design templates for [typography](https://en.wikipedia.org/wiki/Typography), forms, buttons, navigation and other interface components, as well as optional [JavaScript](https://en.wikipedia.org/wiki/JavaScript) extensions. Unlike many web frameworks, it concerns itself with [front-end development](https://en.wikipedia.org/wiki/Front-end_web_development) only.

**5.6 HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**5.7 CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

**CHAPTER-6**

**SYSTEM REQUIREMENTS**

**6.1Hardware Requirements:**

* System    :   Pentium IV 2.4 GHz or above.
* Resolution : Minimum 800X600
* Hard Disk  :   Minimum 40 GB.
* Monitor   :   15 VGA Color.
* RAM    :   Minimum 256 MB.

**6.2Software Requirements:**

* Operating system   : Windows XP, Windows 8/8.1, Windows 10.
* RAM    :   Minimum 4GB
* IDE : Visual Studio 2012
* Database Server : SQL server
* Web Server : Apache Tomcat 7.0

**6.3Functional Requirement**

Following is a list of functionalities of the system. This software should provide the users with the conveniences such as:

* Sign-up and create a new account.
* Log-in using the login-page of the portal.
* Access the skill.
* Able to access some general information related to the skill.
* Able to manage user profiles.
* Learn skill
* Admin is able to delete existing skills
* Admin is able to update new skills
* Admin is able to upload papers from Examination Portal
* Able to see result about the test to the admin as well as user
* Receive mails for any modifications in exam schedule.
* Able to provide results to the user
* Add study materials

**6.4Non-functional Requirement**

* AVAILABLITY - This application is available offline.
* RELIABLITY - This application works same according to documentation.
* FLEXIBLITY - This application works on and above every platform
* SECURITY -This application authenticates user and admin credentials to login to use the application.

**CHAPTER-7**

**DESIGN**

**7.1 MODULE**

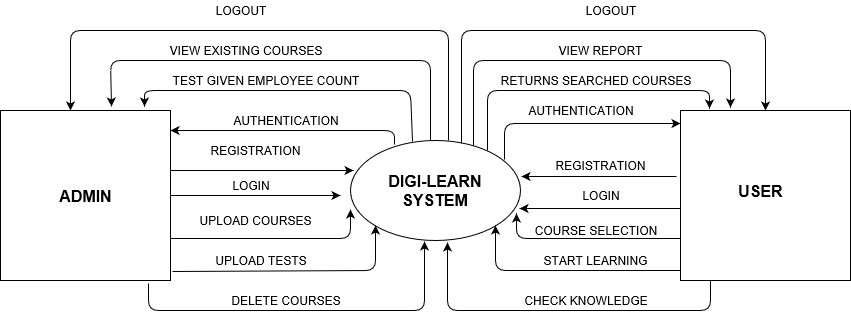
**7.1.1 Module 1: As a User**

|  |
| --- |
| As a user I should be able to sign-up and create a new account so I can avail the services of the portal using the sign-up page. |
| As a user I should be able to log-in using the login-page of the portal so I can access my account. |
| As a user I should be able to access the skills so I that I can learn for the exams I am interested in. |
| As a user I should be able to access some general information related to the skills so that I can be aware of the information. |
| As a user I should be able to apply for exams so I can eliminate the manual process of applying. |
| As a user I should be able to access my result so I don't have to rely on manual methods. |

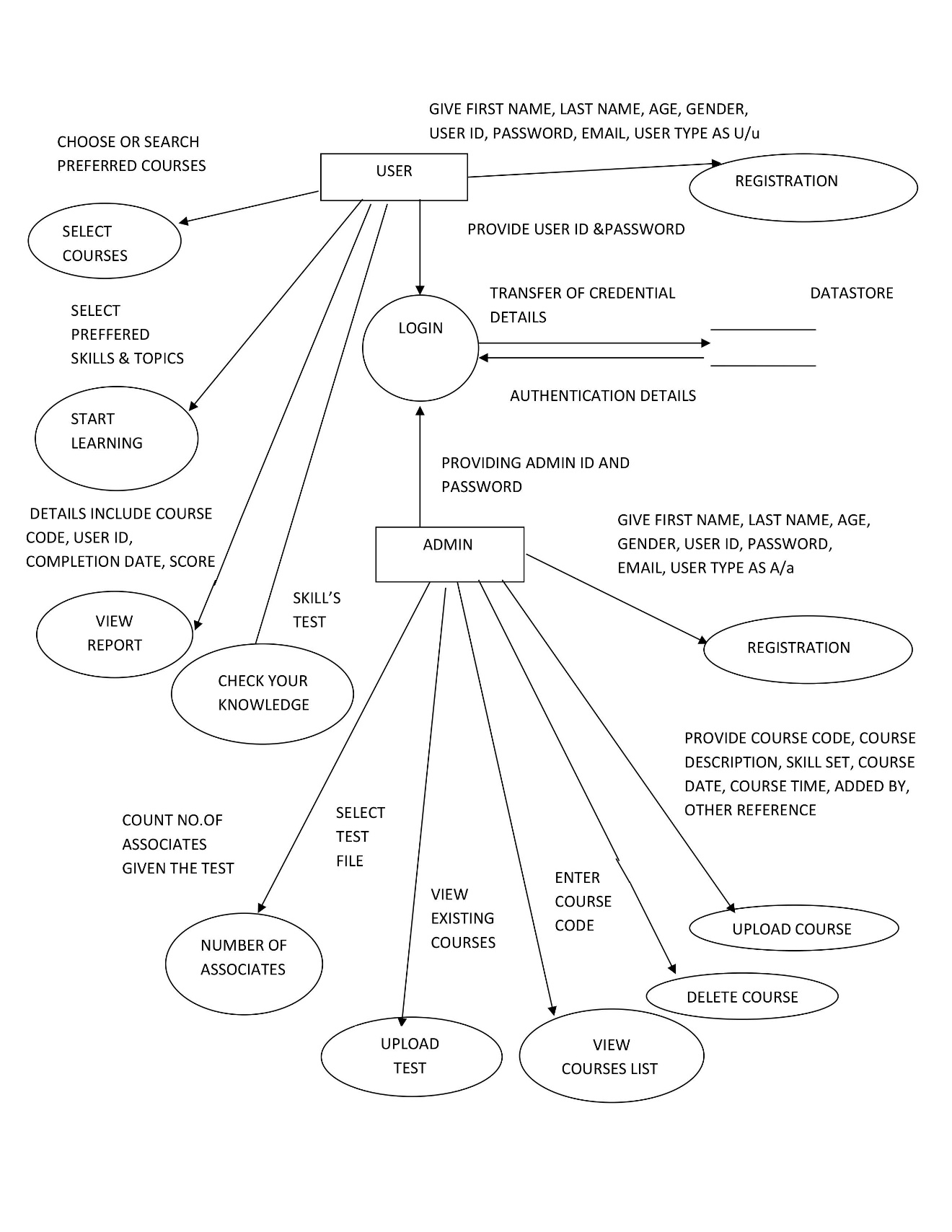
**7.1.2 Module 2: As an Admin**

|  |
| --- |
| As an administrator I should be able to log in with my credentials so I can administrate the portal. |
| As an administrator I should be able to add/modify the skills and lesson so the registered users are updated of the changes. |
| As an administrator I should be able to add/modify the question paper which will be displayed on the portal's homepage and is accessible by user who studies the related course. |
| As an administrator I should be able to add modify delete user and should have access to the details of each user. |
| As an administrator I should be able to provide results to the user so they can access it and skip the delay in accessing the manual result. |
| As an administrator I should be able to add study materials on the Examination Portal |

# **7.2 DATA FLOW DIAGRAM**

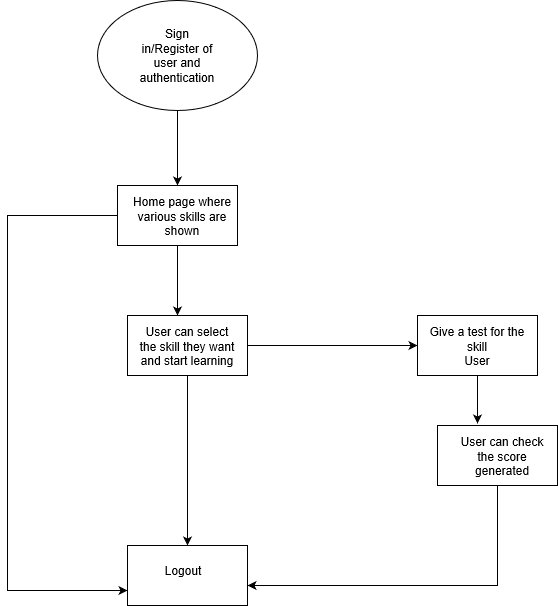


**FIGURE 7.2.1 0-LEVEL DFD**

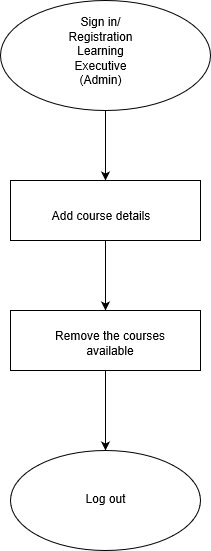
****

**FIGURE 7.2.2 1-LEVEL DFD**

# **7.3 FLOW CHARTS**

****

**FIGURE 7.3.1 USER FLOWCHART**

****

**FIGURE 7.3.2 ADMIN FLOWCHART**

**CHAPTER-8**

**8.1 FRONT END SNAPSHOTS**

****

Fig.8.1.1 Home Page

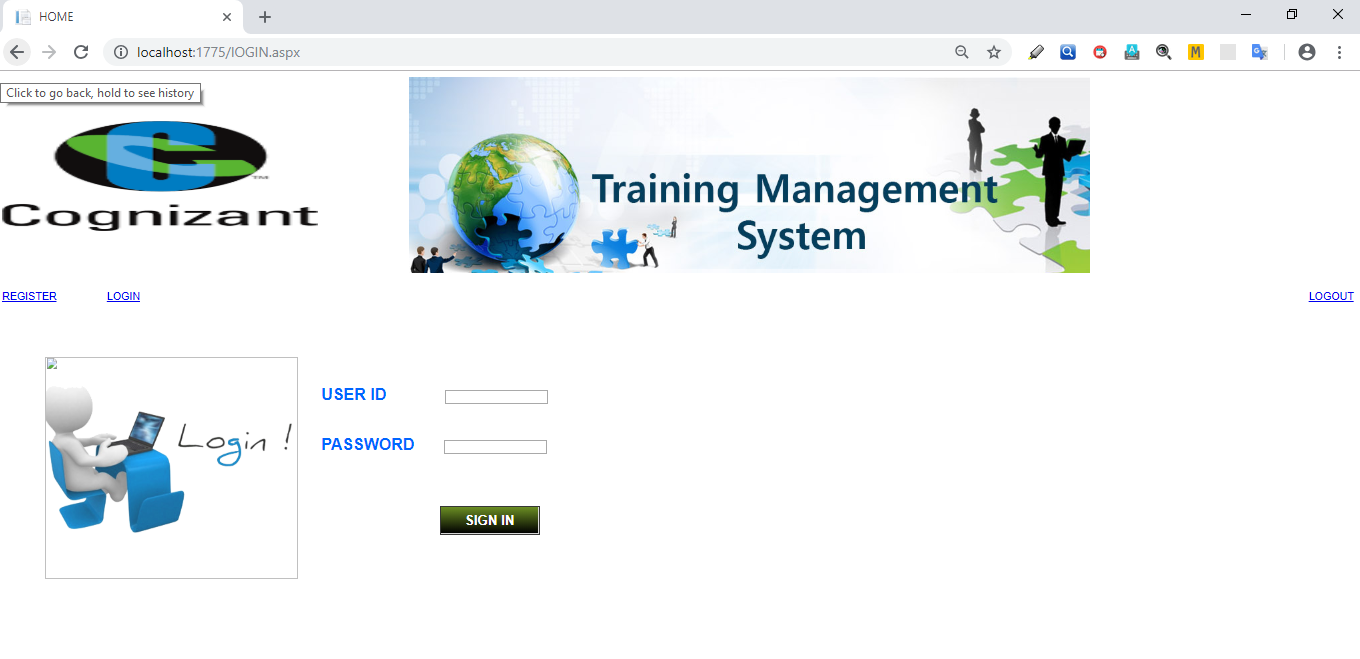


Fig.8.1.2 Login Page

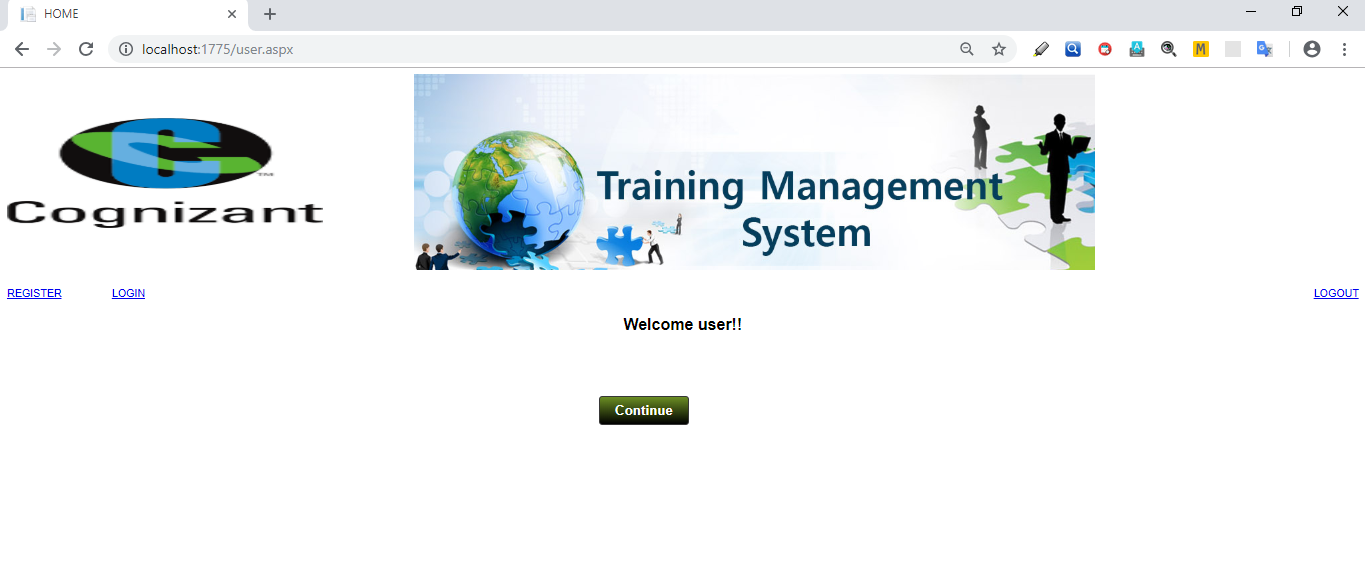


Fig.8.1.3.User Home Page



Fig.8.1.4. Admin Home Page

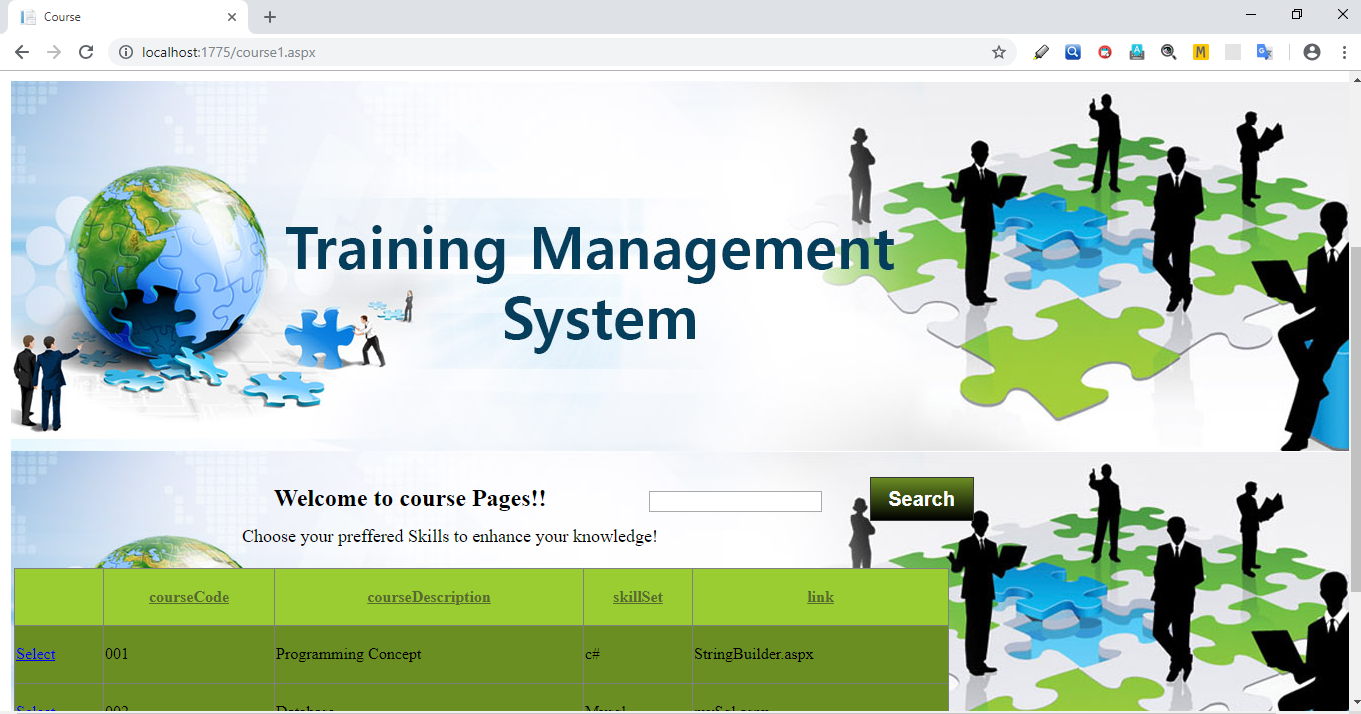


Fig.8.1.5 User Course Page

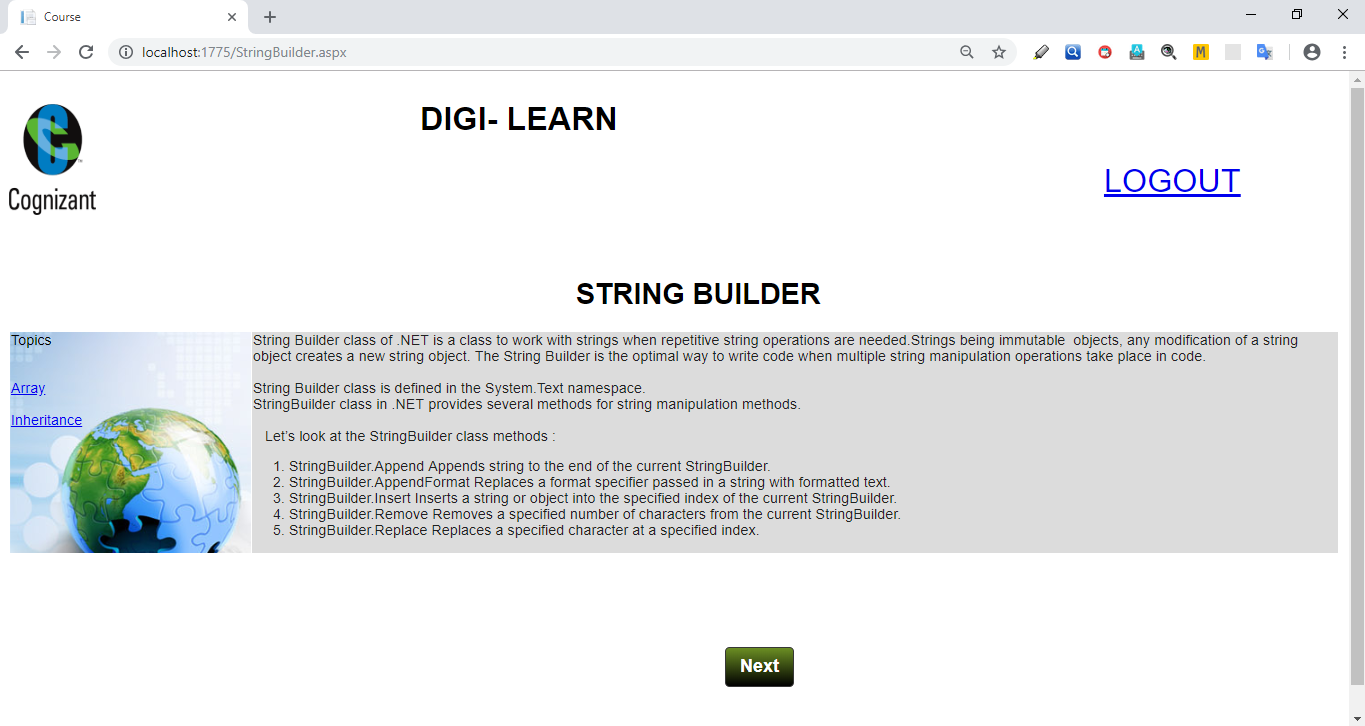


Fig.8.1.6 Study Page

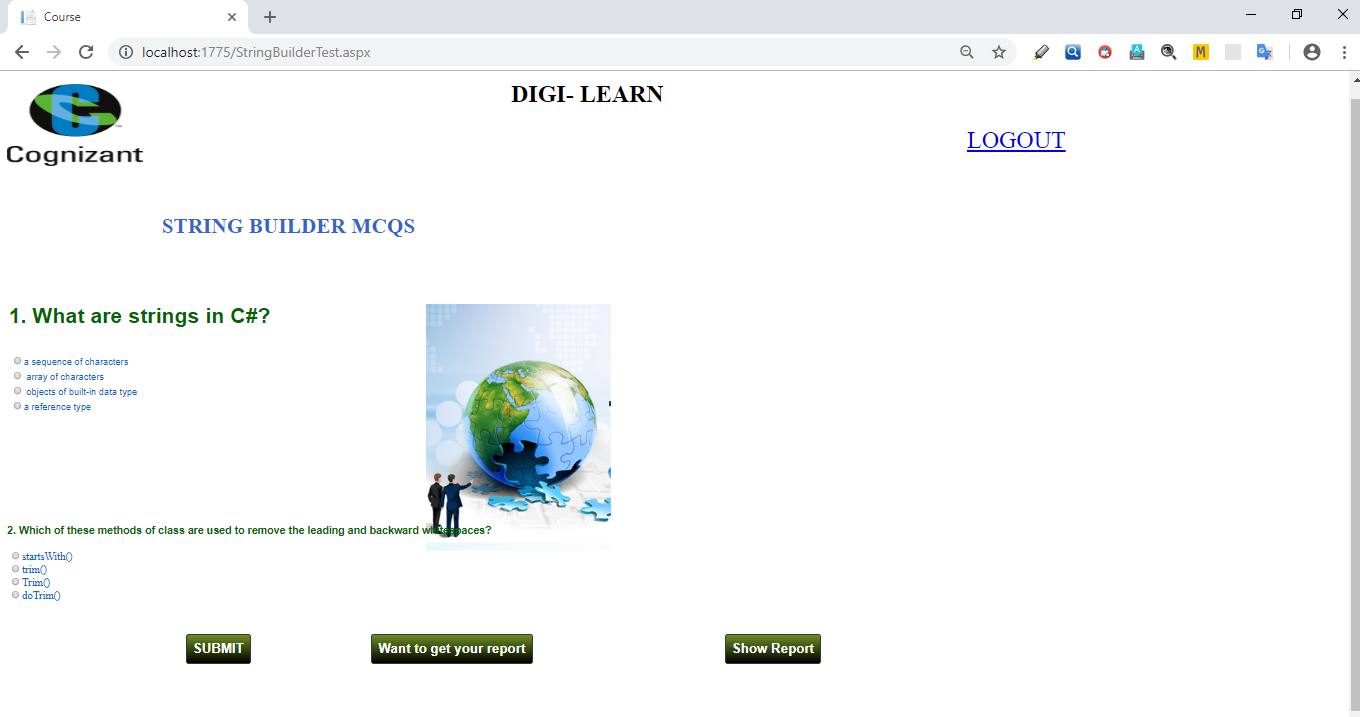


Fig.8.1.7Test Page

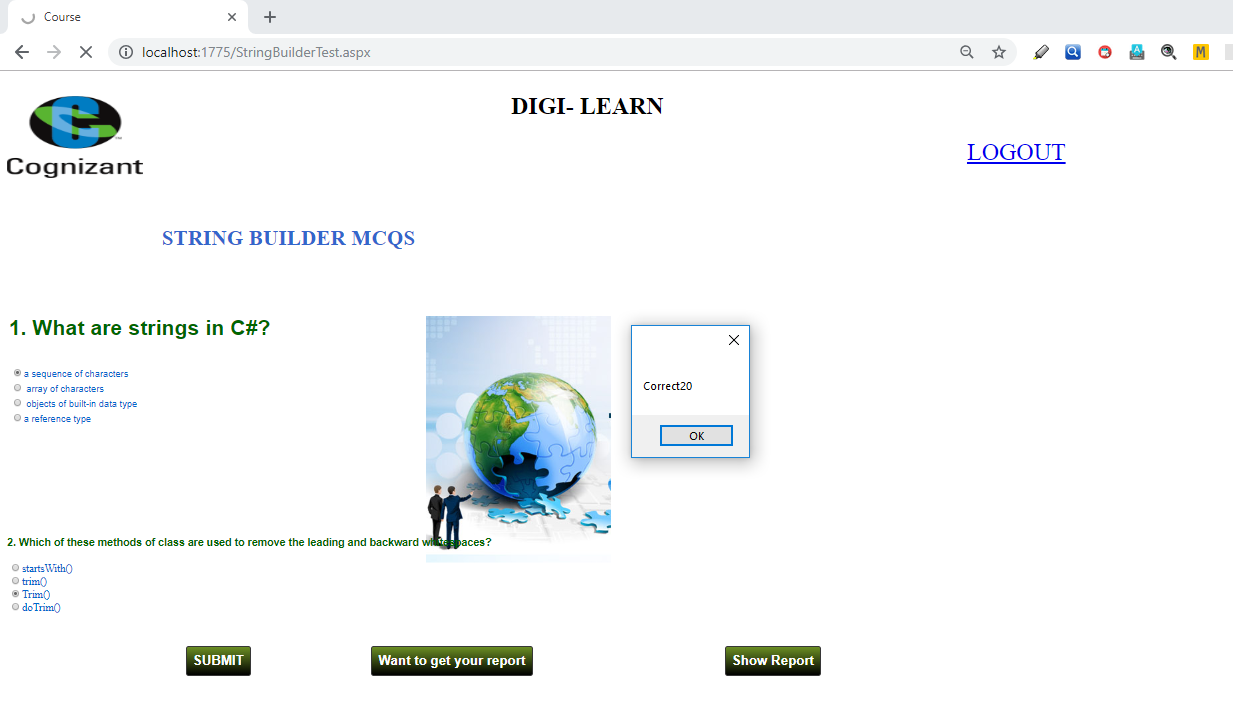
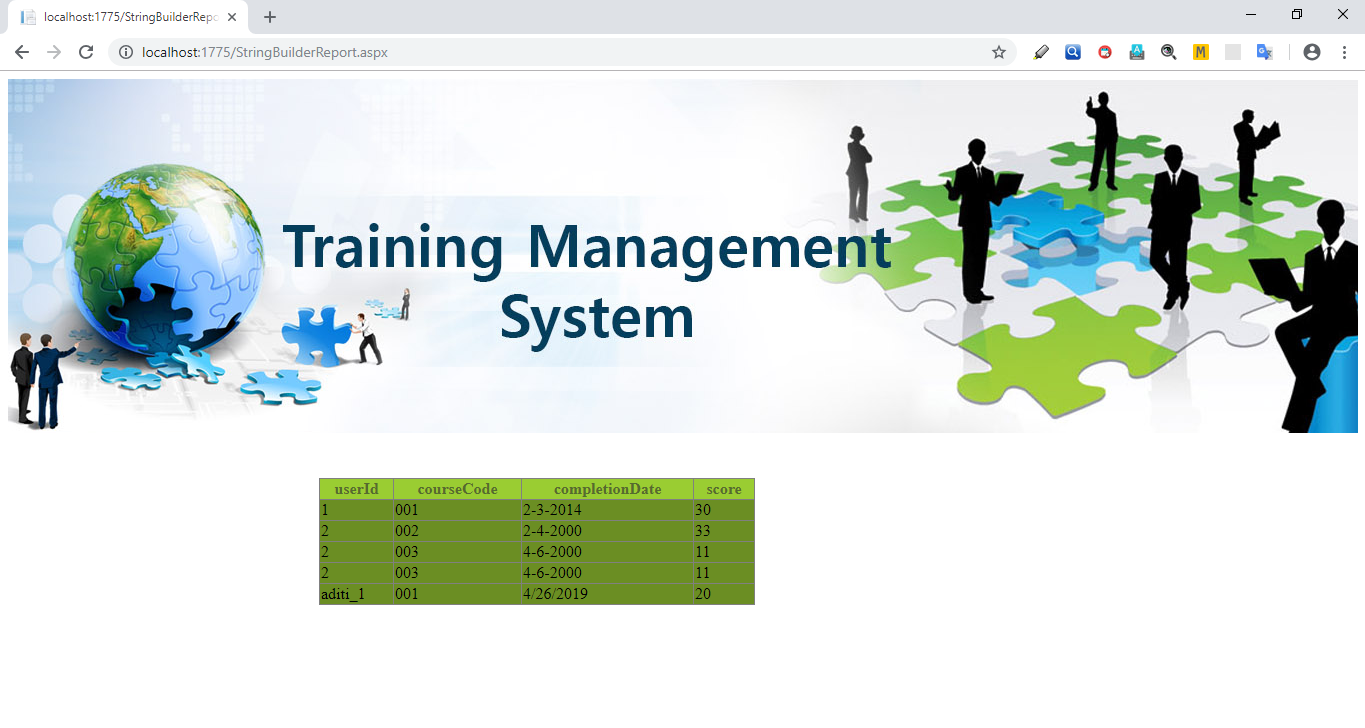


Fig.8.1.8 Test Score

Fig.8.1.9 Report generated for user

**8.2 BACK END SNAPSHOTS**

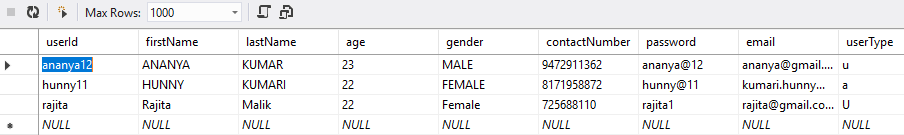


Fig.8.2.1 Registration Table

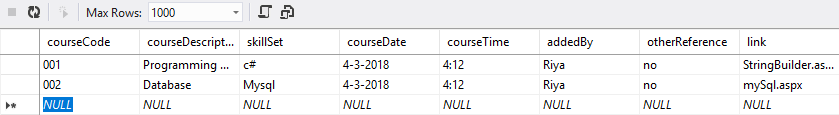


Fig.8.2.2 Course Table

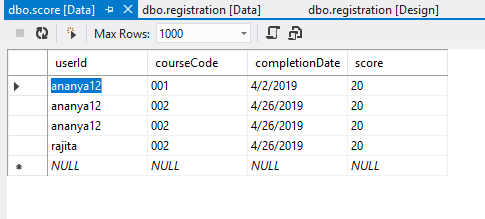
****

Fig.8.2.2 Score Table

**CHAPTER-8**

**TESTING & DEBUGGING**

**8.1 Introduction**

The development of software system involves a series of production activities, where the opportunity for the injection of human fallibilities is enormous. Errors may begin to occur at the very inception of the process of where the objectives may be erroneously on imperfectly specified as well as later design and development stages. Because of human inability to perform and communicate with perfection, software is accompanied by quality assurance activity.

In many ways testing is an individual elastic process and the number of different types of lists varies is much as the different development approaches. For many years, our only defense against programming errors was careful design and the native and intelligence of the programmer. We are now in era in which modern design techniques enable us to reduce the number of initial errors that are inherent in the code.

Similarly, different test methods are beginning to cluster themselves into several distinct approaches and philosophies. Testing is a critical phase in system implementation. Testing of system involves hardware devices testing and debugging of computer programs and testing information processing procedures. Testing can be done with the test data which attempts to simulate all possible condition that may arise during processing. If Structured programming methodologies have been adopted during coding, the testing proceeds from the higher to the lower level of programming modules until entire program is tested.

**CHAPTER-9**

**TYPES OF TESTING**

**9.1 Black Box Testing**

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

**9.2 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.

**9.3 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 driver modules or test harnesses.

**9.4 Incremental Integration Testing**

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

**9.5 Integration testing**

Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**9.6 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.

**9.7 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.

**9.8 End To End Testing**

Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

**9.9 Regression Testing**

Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**9.10Acceptance Testing**

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

**9.11 Load Testing**

It is a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

**9.12 Test Cases**

The primary objective of test case design is to derive a set of tests that have the highest likelihood of uncovering errors in software. The test case specification is the major activity in the testing process. Careful selection of test cases that satisfy the criterion on approach specified is essential for proper testing. Various characteristics of test cases that are required for portal are:

A good test has a high probability of finding an error. A good test is not redundant. A good test should be “Best of Breed”. A good test should be neither too simple not too complex.

Test Data for This Application:

1. User Details like Name, Employee-ID, etc.

2. Project Details like hands-On, objective, subject-wise, exam-wise, etc.

**CHAPTER-10**

**CONCLUSION AND FUTURE WORK**

**CONCLUSION**

Such an application created which is concise and supported in every operating system. Thus there is scope for backward compatibility.

The web-app minimizes resource consumption on the system and aims to be efficient and fast.

To improve the user friendliness the app uses a simple user interface, thus ensuring maximum possible accessibility.

**SCOPE OF FUTURE WORK**

There are some features which can be implemented further.

* Redesigning the Database Structure and making it more advanced.
* With a bigger team the project can be implemented using advanced JavaScript frameworks.
* Increasing the scalability of the application so that many users can use it simultaneously.
* We can maintain an attendance record of a trainee during training.

**REFERENCES**

[1] [www.stackoverflow.com](http://www.stackoverflow.com/)

[2] [www.csharptutorial.com](http://www.csharptutorial.com)

[3] [www.tutorialspoint.com](http://www.tutorialspoint.com)

[4] www.w3schools.com

[5] [www.github.com](file:///C:\Users\A%20K%20Shrivastav\Desktop\www.github.com)

[6]http://info.dogpile.com/dogpl.305.10/search/web?fcoid=417&fcop=topnav&fpid=2&cid=319942383&ad.segment=dogpl.305.10&ad.device=c&aid=a477e374-d859-4803-8b88-c87f836344b9&ridx=1&q=e+survey+functional+requirements&ql=&ss=t

[7] "Supplement 4-A, A Procedure for Requirements Analysis". [Systems Engineering Fundamentals](http://www.dau.mil/publications/publicationsDocs/SEFGuide%2001-01.pdf) (PDF). United States Government US Army. 2001. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [978-1484120835](https://en.wikipedia.org/wiki/Special:BookSources/978-1484120835). Retrieved 18 March 2016.

[8] Roman, G.-C. (1985) "A taxonomy of current issues in requirements engineering" Computer, April 1985 Volume: 18 Issue: 4, pp. 14–23, [ISSN](https://en.wikipedia.org/wiki/International_Standard_Serial_Number) [0018-9162](https://www.worldcat.org/search?fq=x0:jrnl&q=n2:0018-9162).

[9] R. T. Yeh (1982) "Requirements Analysis- A Management Perspective," Proc. COMPSAC '82, Nov. 1982, pp. 410–416.

[10] Wiegers, Karl E. (2003). [Software Requirements](http://processimpact.com/pubs.shtml#reqs) (2nd ed.). Redmond: Microsoft Press. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [978-0-7356-1879-4](https://en.wikipedia.org/wiki/Special:BookSources/978-0-7356-1879-4).

[11] Stellman, Andrew; Greene, Jennifer (2005). [Applied Software Project Management](http://www.stellman-greene.com/) (1st ed.). Cambridge, MA: O'Reilly Media. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [978-0-596-00948-9](https://en.wikipedia.org/wiki/Special:BookSources/978-0-596-00948-9).

[12][Sommerville, Ian](https://en.wikipedia.org/wiki/Ian_Sommerville_(academic)" \o "Ian Sommerville (academic)) (2006). [Software Engineering](http://www.cs.st-andrews.ac.uk/~ifs/Books/SE8/index.html) (8th ed.). [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [978-0-321-31379-9](https://en.wikipedia.org/wiki/Special:BookSources/978-0-321-31379-9).

The Complete Reference JAVA – Herbert Schildt