**ABSTRACT**

Infosys focuses on online training for employees across DC’s. The ETA heads need a system that can automatically record and store course-wise session recording which can be referred to by any employees at any point of time. Every online course should be supplemented with the corresponding question bank. After completing a course, the learner must attempt the test in order to achieve a badge - Completed Course.

Any employee interested to undergo training should be able to register himself/herself with the system for the desired course. The employee should be able to log in to the system, view and update subjects of interest for selfpaced training and attend appropriate session as convenient.

# ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgement, then let words play the heralding role in expressing my gratitude. To bring something into existence is truly a work of God. I would like to thank God for not letting me down and showing me the silver lining in the dark clouds.

I would like to thank Dr. M D Mathew, Principal, Saintgits College of Engineering for his support and encouragement. I convey my heartfelt thanks to Dr. Rajesh K.S (Head of the Department of Master of Computer Applications, Saintgits College of Engineering) for providing an opportunity for the project presentation. It is my pleasure to express my gratitude to the project coordinator Asst.Prof. Libin M Joseph, Department of Computer Applications, Saintgits College of Engineering whose support and constructive criticism has led to the successful completion of the task.

With the biggest contribution to this report, I would like to thank Mr.Gopeekrishnan R, Department of Computer Applications who had given me full support in guiding me with stimulating suggestions and encouragement to go ahead in all the time of the this work. I would also thank my institution and faculty, my family and friends without whom this project would have been a distant reality.

**Devika M Adhithyu**

**CONTENTS**

# Title Page No.

**CHAPTER 1**

**INTRODUCTION**

* 1. **PROBLEM DEFINITION**
  2. **ABOUT THE ORGANIZATION**

Infosys was established by seven engineers in Pune, Maharashtra, India with an initial capital of $250 in 1981. It was registered as Infosys Consultants Private Limited on 2 July 1981. In 1983, it relocated its office to Bangalore, Karnataka, India.

Infosys Ltd is a global technology services firm that defines, designs and delivers information technology (IT)-enabled business solutions to their clients. The company provides end-to-end business solutions that leverage technology for their clients, including technical consulting, design, development, product engineering, maintenance, systems integration, package-enabled consulting, and implementation and infrastructure management services.

The company also provides software products to the banking industry. They have developed Finacle, a universal banking solution to large and medium size banks across India and overseas. Infosys BPO is a majority owned subsidiary. Through Infosys BPO, the company provides business process management services, such as offsite customer relationship management, finance and accounting, and administration and sales order processing. The company is having marketing and technical alliance with FileNet, IBM, Intel, Microsoft, Oracle and System Application Products.

Infosys Ltd is a public limited and India's second largest software exporter company was incorporated in the year 1981 as Infosys Consultants Pvt Ltd by Mr. N.R. Narayana Murthy at Karnataka. The company was started by seven people with the investment of USD 250. The company became a public limited company in the year 1992. The company was the first Indian company to be listed on the NASDAQ in the year 1999. Infosys also forms a part of the NASDAQ-100 index. Continuously in the year 2001, 2002 and 2003, the company wins the National award for excellence in corporate governance conferred by the Government of India.

* 1. **OBJECTIVE OF THE PROJECT**

Selfpaced Learning is an online website which is mainly focuses on employees in Infosys DC’s. The ETA heads need a system that can automatically record and store course-wise session recording which can be referred to by any employees at any point of time. Every online course should be supplemented with the corresponding question bank. After completing a course, the learner must attempt the test in order to achieve a badge-Completed Course.

Using this website an employee can register and login to the website. After login he can start studying the languages as per his/her needs. After completing a particular course the employee can take test and he can score marks. If the employee performed well in the examination then a ranking is allotted to the employee. After completing the course badges is allotted to the employee. For each completed course the employee have that number of badges. The employee can view this badges from his/her profile. After completing a course the employee can generate certificate for that particular course . He can download this certificate from the website .

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 INITIAL INVESTIGATION**

**2.2 EXISTING SYSTEM**

Many Education website have been developed based upon different platforms and concepts. Use of education related apps is growing but there are many issues related to their functionality. “ILP” is a Information Learning Platform for Infosys employees in DC’s. By using the ILP platform employees in Infosys can learn different courses . Employees can study different courses as per their needs. After completing a course employee can take a test . During the courses they can attempt quizzes.

**2.3 PROPOSED SYSTEM**

Selfpaced Learning is an online website which is mainly focuses on employees in Infosys DC’s. The ETA heads need a system that can automatically record and store course-wise session recording which can be referred to by any employees at any point of time. Every online course should be supplemented with the corresponding question bank. After completing a course, the learner must attempt the test in order to achieve a badge-Completed Course. Certification is also one of the best advantage of the proposed system. After completing the course employee can request to generate certificate and employee can download the certificate.

**2.3.1 User Classes and Characteristics**

The system contains two users: Admin,Employee. Admin deals with adding and editing certification details, giving permission to a set of employees for certification etc. Employee deals with selection of course and rules to complete the courses are configured. Through these functionalities the proposed work tries to overcome all the listed disadvantages of existing systems.

**2.4 FEASIBILITY STUDY**

During system analysis, a feasibility study of the proposed system was carried out to see whether it was beneficial to the Educational field. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product. While evaluating the existing system, many advantages and disadvantages raised. Analysing the problem thoroughly forms the vital part of the system buddy. Problematic areas are identified and information is collected.

The benefits of this site are users can easily interact and get the services without much complexity. It helps to make it possible that more users can interact with the site at a time. Feasibility study is to determine whether the proposed system is technically, economically and behaviorally feasible in all respects.

The main aim of feasibility study is to evaluate alternative site and propose the most feasible and desirable site for development. If there is no loss for the organization then the proposed system is considered financially feasible. A feasibility study is carried out to select the best system that meets performance requirements. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system.

In this scenario, problems are identified. Essential data are being gathered for the existing problems. It is necessary that this analysis familiarizes the designer with objectives, activities, and the function of the organization in which the system is to be implemented. The feasibility study was divided into four :-

Technical, Economical, Operational and Behavioral.

It is summarized below :-

**2.4.1 Technical Feasibility**

According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs, are identified. While considering the problems of existing system, it is sufficient to implement the new system. The proposed system can be implemented to solve issues in the existing system. It includes the evaluation of and how it meets the proposed system.

**2.4.2 Economical Feasibility**

Economic analysis is most frequent used for evaluating of the effectiveness of the candidate system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a candidate system and compare them with the existing system. Except for the initial capital amount and the amount after each financial year, no other huge amount is needed. The expenses can be handles by any participants. So, the system is economically feasible.

This feasibility involves some questions such as whether the firm can afford to build the system, whether its benefits should substantially exceed its costs, and whether the project has higher priority and profits than other projects that might use the same re- sources. Here there is no problem. This firm has fully equipped hard ware, and fully fledged software, so no need to spend money on these issues. And as the client and the developer are one, there is no further problem in economic issues.

**2.4.3 Operational Feasibility**

Methods of processing and presentation are all according to the needs of clients since they can meet all user requirements here. The proposed system will not cause any problem under any circumstances and will work according to the specifications mentioned. Hence the proposed system is operationally feasible. People are inherently resistant to change and computer has been known to facilitate changes. The system operation is the longest phase in the development life cycle of a system. So, Operational Feasibility should be given much importance. This system has a user-friendly interface. Thus it is easy to handle.

**2.4.2 Behavioral Feasibility**

In today’s world, computer is an inevitable entity. As per the definition of behavior design, many valid points are recognized in this study. This system behavior changes according to different environment. In order to ensure proper authentication and authorization and security of sensitive data of the admin or employers, login facilities are provided. These are the main feasibility studies tested in this application**.**

**CHAPTER 3**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 SOFTWARE REQUIRMENT AND SPECIFICATION**

Selfpaced Learning is an online website which is mainly focuses on employees in Infosys DC’s. The ETA heads need a system that can automatically record and store course-wise session recording which can be referred to by any employees at any point of time. Every online course should be supplemented with the corresponding question bank. After completing a course, the learner must attempt the test in order to achieve a badge -Completed Course. Certification is also one of the best advantage of the proposed system. After completing the course employee can request to generate certificate and employee can download the certificate.

**Module Description**

**Admin:** This module deals with adding and editing certification details, giving permissions to a set of employees for certification etc.

* An admin should be able to approve details of employees if they add any extra information or details about them in their profile by properly verifying it.
* Should be able to view the number of employees opted for particular project.
* Should be able to book an appointment for particular employee through online web interaction and for other complete details.
* Should be able to view the feedback report given by the employees.
* Should be able to export all employee related data .csv file format.

**Employee:** In this module, selection of course and rule to complete course are configured.

* Employees can sign up by providing their details of the proficiency in a programming language.
* Signed employees can login and check for the certification that are suitable for their area of interest.
* Employees can give the feedback about the certification that have under taken, its level of complexity and deadline time assigned.

**3.1.1 FUNCTIONAL REQUIRMENTS**

**3.1.2 NON FUNCTIONAL REQUIRMENTS**

**3.2 UML DIAGRAM**

UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar JAcobson and the Rational Software Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. Today, UML is accepted by the Object Management Group(OMG) as the standard for modelling software development.

UML stands for Unified Modeling Language. UML 2.0 helps extend the original UML specification to cover a wider portion of software development efforts including agile practices. Improved integration between structural models like class diagrams and behavior models like activity diagrams. The original UML specified nine diagrams; UML

* 1. brings that number up to 13. The four new diagrams are called: communication diagram, composite diagram, interaction overview diagram and timing diagram. It also renamed state chart diagrams to state machine diagrams , also known as state diagrams.

## Types of UML diagrams

The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing and deployment. These diagrams are organized into two distinct groups: structural diagrams and behavioral or interaction diagram.

## Structural UML diagrams

* Class diagram
* Package diagram
* Object diagram
* Component diagram
* Composite structure diagram
* Deployment diagram

## Behavioral UML diagrams

* Activity Diagram
* Sequence diagram
* Use case diagram
* State diagram
* Communication diagram
* Interaction overview diagram
* Timing diagram

## Usercase Diagram

To model a system the most important aspect is capture the dynamic behaviour. To modify a bit in details, dynamic behaviour of the system when it is running or operating. So only behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagram consists of actors, use case and their relationships. The diagram is used to model the system of an application. A single use case diagram captures a particular functionality of a system.

**Use case Diagram objects:**

* Actor
* Use case
* System
* Package

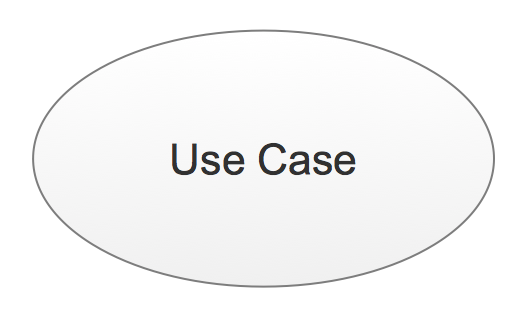
## Actor

Actor is a use case diagram in an entity that performs a role in one given system. This could be a person, organization or an external system usually drawn like skeleton.

**Fig1:Actor**

**Use case**

A use case represents a function or an action within the system. Its drawn as an oval and named with the function.

****

**Fig2:Use Case**

## System

System is used to define the scope of the use case and drawn as a rectangle. This is an optional element but useful when your visualizing large systems. For example you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.

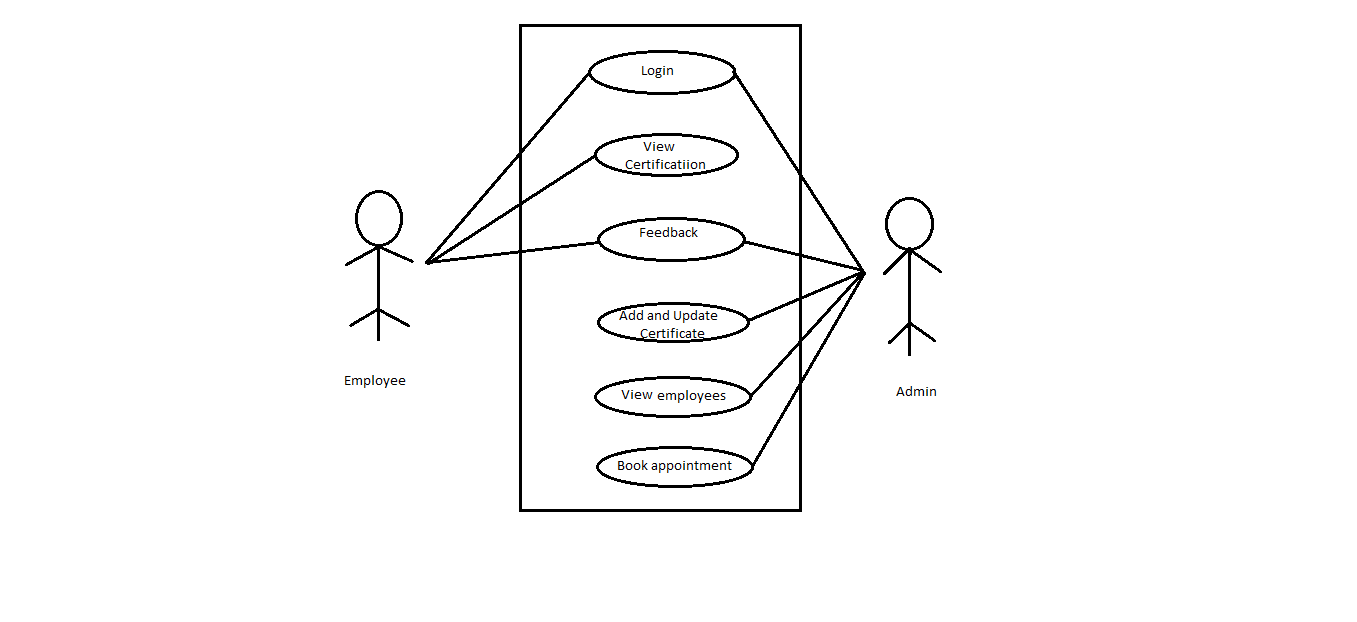
## Package

Package is another optional element that is extremely useful in complex diagrams.

Similar to use class diagrams, packages are used to group together use cases.

* + 1. **Sequence Diagram**

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system.Time is represented in the vertical direction showing the sequence of interaction of the header elements.Sequence Diagrams are used primarily to design, document and validate the architec ture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.



Although UML sequence diagrams are typically used to describe object-oriented software systems, they are also extremely useful as system engineering tools to design system architectures in business process, as message sequence charts and call flows for telecoms or wireless system design, and for protocol stack design and analysis.

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence Diagrams are typically associated with use case realizations in the logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenar- ios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

**3.3 SYSTEM DESIGN**

Design is the abstraction of a solution; it is a general description of the solution to a problem without the details. Design is view patterns seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required to cre- ate the implementation. The design is a solution, the transition of requirements in two ways of meeting them. The design will determine the success of the system. Based on the proposed system objectives, the major modules are identified and the operations to be carried out are determined. In the design phase of the system the user interaction screen, database tables ,inputs, outputs and screen are designed. The database tables are designed by using all the necessary fields in a compact manner. The redundancy and duplication of fields are avoided.

Systems design involves first logical design and the n physical construction of the system. After logical design, a detailed specification of the system, which describes the inputs, outputs, files are developed. During the design phase of system the following factors are considered.

The important of software design can be stated with a single word quality. Design is placed where quality is fostered in software development. Design is the only way whose requirements are actually translated into a finished software product or system.

* Data Floors:-The movement of data into, around and out of the system
* Data Source:-Temporary and permanent collections of data
* Processors:-Activities to accept, manipulate and deliver data and information
* Procedures:-Methods and routines to achieve the intended results

## Input Design

The input is the set of values that is provided by the user to the system.The input design must enable the user to provide the error free input to the system for efficient pro- cessing.The data is fed into the system using simple interactive xml pages.The pages have been supplied with messages so that user can enter data without facing any diffi- culty.The data is validated wherever it requires in the project

The main objectives of the input design are as follows:

* Produce effective method of input
* Achieve high level accuracy
* Ensure that the input is acceptable and understood by the user .

The different types of input data handled by the system are:

## External

They are the primary inputs to the system. The external input is what the user supplies to the system. The user can give different types of external inputs in this project such as registration details ,login details etc.

## Internal

When the external inputs are obtained from the user, these inputs are transferred to the system as messages. These messages are captured and handled as input for further processing.

In this project the input design is done with Android and PHP codes. The external in- puts are the data given to the system by the user. The neccessary external inputs are given to the system by Graphical User Interface(GUI)technology. The GUI system applied to this project enables the user to avoid error and confusion arises while entering the input.

## Output Design

The primary consideration in the design of all output is the information requirement and other objective of the users. It is the most important and direct source of information to the user. A major form of output is a hard copy. Print out should be designed around the output requirements of the user. Each output should be given a specific name or title. The output data is displayed on the visual display unit and output can be redirected to printers and or sorted in a file for later use.

## Database Design

Database is a design to manage large bodies of information. The management of data involves both the definition of structures for the storage information. In addition, the database system must provide for the safety of the information solved, despite system crashes or due to attempts at unauthorized access. For developing an efficient database we have to fulfil certain condition such as controlled redundancy

* Defining the data
* Inputting the data
* Locating the data
* Accessing the data
* Communicating the data
* Revising the data

**OBJECTIVES OF DATABASE** In the database design, several objectives are considered such as

* Control of data Integrity
* Ease of use
* Control of redundancy
* Control of security
* Data independence(Logical and physical)
* Data storage protection(Record level and Table level)
* System performance
* System functions
* System compatibility

For achieving the above mentioned criteria we have to make use various features that are available with the RDBMS by enforcing integrity constraints, we can ensure data integrity and reduce data inconsistency to a great extent. Recovery from failures can overcome using backup facilities. By using table level as well as row level locking facilities , we can avoid concurrent access normalize. Another important features of RDBMS is the logical and physical data independence. In addition to security mechanism provided by RDBMS, we have provided system password to near system

**NORMALIZATION:** Normalization is the term obtained from the Latin word NORMA which means that square used by the carpenter . Normalization is the process of simplifying the relationship between data elements in a record, through normalization a collection of data I a record structure is replaced by successive record structures that

are simpler and can be managed efficiently. While designing the database, we have to implement the concept of normalization to avoid data redundancy in the database. Normalization is carried out for four reasons.

* To structure the data so that any pertinent relationship between entities can be represented.
* To permit simple retrieval of data in response to query and reports required.
* To simplify data maintenance procedures such as insertion, deletion and updating.
* To reduce the need to be structure or reorganize data with new application requirements arise.

The major normalization strategies are

* First Normal Form
* Second Normal Form
* Third Normal Form
* Boyce/Codd Normal Form(BCNF)

**FIRST NORMAL FORM:** First Normal Form is achieved when all repeating groups in a record are removed, so that record is of fixed length. A repeating group, reoccurrence of a data item or group of data item within a record indicates another relation.

**SECOND NORMAL FORM:** Second Normal Form is achieved when a record is in first normal form and each item in the record is functionally depend on the primary key for identification. In other words, analyst seeks functional dependency. A data item is functionally dependent of its value is uniquely associated with a specific data item is functionally dependent of its value is uniquely associated with a specific item. To achieve second normal form every column in a table that is not dependent on the primary key of the record should be removed and used to form a separate relation.

**THIRD NORMAL FORM:** Third Normal Form is achieved when all transitive dependencies are removed from a record. That is, if A is functionally dependent on B and B is functionally dependent on C, then A is functionally dependent on C.

**BOYCE/CODD NORMAL FORM(BCNF):** BCNF is often used to distinguish the new 3NF from the old. An attribute possible composite is called as determinant. If other attributes are fully functionally determined this attribute(or on which some other attribute is fully functionally dependent on this attribute).A table is in BCNF , if every determinant is a candidate key. To achieve a table is in BCNF, remove fields which are fully functionally dependent on a determinant, which is not act as a candidate key.

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| empid | Number(3) | Primary key | It is used to store the unique id of employee |
| name | Varchar(200) | Not  Null | Name of Employee |
| dob | Date |  | Date of birth |
| password | Varchar(500) | Not  Null | Password |
| email | Varchar(200) | Not  Null | Email id |
| ranking | Number(2) |  | Ranking |
| badges | Number(2) |  | Badges |
| phone\_number | Int | Not  Null | Phone number |

**Table: Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| courseid | Number(4) | Primary key | It is used to store the unique id of course |
| no\_of\_enroll | Number(2) | Not  Null | Number of enrollment |
| course\_Name | Varchar(25) |  | Name of the course |
| description | Varchar(40) | Not  Null | Description about the course |
| duration | Number(3) | Not  Null | Duration of the course |

**Table: Course**

**Table: Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| empid | Number(3) | Foreign key  Not Null | It is used to store the unique id of employee |
| courseid | Number(4) | Foreign key  Not Null | It is used to store the unique id of course |
| feedbackid | Number(5) | Primary key | It is used to store the unique id of feedback |
| Review | Varchar(50) | Not  Null | Review |

**Table: EmployeeCourse**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| empcouid | Number(4) | Primary key | It is used to store the unique id of employee courses |
| courseid | Number(4) | Foreign Key | It is used to store the unique id of course |
| empid | Number(3) | Foreign Key | It is used to store the unique id of employee |
| status | Varchar(10) |  | Status |
| trail | Number(3) |  | Number of trials |
| max\_score | Number(5,2) |  | Maximum score |

**Table: EmployeeCourse**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| empid | Number(3) | Foreign key  Not Null | It is used to store the unique id of employee |
| courseid | Number(4) | Foreign key  Not Null | It is used to store the unique id of course |

**3.4 TOOLS AND APPLICATION**

**3.4.1 ANGULAR 4**

UI designing is one of the most crucial parts of any application. UI is where the user interacts with any application, hence it has to be designed in a user-friendly manner. Many frameworks have come into existence for the same. Angular is one such very powerful framework for building client applications using any scripting language such as JS or TypeScript. The most preferred scripting language for Angular is TypeScript which can again be compiled to JavaScript. Any UI designer would appreciate Angular if he already knows any traditional approaches as this framework helps us in designing application with better performance and maintainability for both mobiles and desktops.

AngularJS is based on the model view controller, whereas Angular 2 is based on the components structure. Angular 4 works on the same structure as Angular2 but is faster when compared to Angular2.

Angular4 uses TypeScript 2.2 version whereas Angular 2 uses TypeScript version 1.8. This brings a lot of difference in the performance.

To install Angular 4, the Angular team came up with Angular CLI which eases the installation.

The Angular 4 app folder has the following **folder structure** −

* **e2e** − end to end test folder. Mainly e2e is used for integration testing and helps ensure the application works fine.
* **node\_modules** − The npm package installed is node\_modules. You can open the folder and see the packages available.
* **src** − This folder is where we will work on the project using Angular 4.

The Angular 4 app folder has the following **file structure** −

* **.angular-cli.json** − It basically holds the project name, version of cli, etc.
* **.editorconfig** − This is the config file for the editor.
* **.gitignore** − A .gitignore file should be committed into the repository, in order to share the ignore rules with any other users that clone the repository.
* **karma.conf.js** − This is used for unit testing via the protractor. All the information required for the project is provided in karma.conf.js file.
* **package.json** − The package.json file tells which libraries will be installed into node\_modules when you run npm install.

The src folder is the main folder, which internally has a different file structure.

**app**-It contains the files described below. These files are installed by angular-cli by default.

* app.module.ts − If you open the file, you will see that the code has reference to different libraries, which are imported. Angular-cli has used these default libraries for the import – angular/core, platform-browser. The names itself explain the usage of the libraries.

They are imported and saved into variables such as **declarations, imports, providers**, and **bootstrap**.

**declarations** − In declarations, the reference to the components is stored. The Appcomponent is the default component that is created whenever a new project is initiated. We will learn about creating new components in a different section.

**imports** − This will have the modules imported as shown above. At present, BrowserModule is part of the imports which is imported from @angular/platform-browser.

**providers** − This will have reference to the services created. The service will be discussed in a subsequent chapter.

**bootstrap** − This has reference to the default component created, i.e., AppComponent.

* **app.component.css** − You can write your css structure over here. Right now, we have added the background color to the div as shown below.
* **app.component.html** − The html code will be available in this file.

This is the default html code currently available with the project creation.

* **app.component.spec.ts** − These are automatically generated files which contain unit tests for source component.
* **app.component.ts** − The class for the component is defined over here. You can do the processing of the html structure in the .ts file. The processing will include activities such as connecting to the database, interacting with other components, routing, services, etc.

**Angular and TypeScript**

Angular is one of the most powerful client-side U.I. framework which can be used to develop **complex**, **customizable**, **modern**, **responsive** and **userfriendly** web applications. Some such applications are PayPal, Netflix, Weather etc.Angular is a single framework which addresses concernsof bothmobileanddesktop application.The Angular framework has chosen a **new statically-typed**, **client-side scripting** language called TypeScript, which beautifully works around most if not all such pitfalls of JavaScript. The Angular team recommends the usage of TypeScript for Angular applications. Hence, to start building Angular applications, we would need to learn to write simple TypeScriptcode first and then look at the Angular application design.

**Pitfalls of JavaScript**

JavaScript is the language used for client side scripting. We can do client side validations, DOM manipulation, Ajax calls etc using JavaScript. We can also use JavaScript frameworks for writing complex business logic which runs on the client side.

As the complexity of the JavaScript code increases, it gradually becomes difficult in coding and maintaining. This is because of the following pitfalls of JavaScript language.

**Dynamic Typing**: Dynamic typing means deciding the data type of the variable dynamically at runtime which results in recompilation every time the code is executed.

**Interpreted Language**: Interpreted Language is a language in which the code instructions are executed directly without prior compilation to machine-language instructions hence we will not get to know the errors until the code is executed.

**Minimal Object Oriented support**: JS supports minimal Object Oriented concepts like classes, encapsulation, inheritance which helps in readability and reusability of the code.

**Minimal IDE support**: Integrated development environment (IDE) is a software application that provides all necessary options like code refactoring, intellisense support, Debugging support to software programmers for software development which is least supported in JS.

Albeit all these shortcomings, we are still dependent on JS as it is the most common scripting language for browsers. But as programmers, we might be in luck.

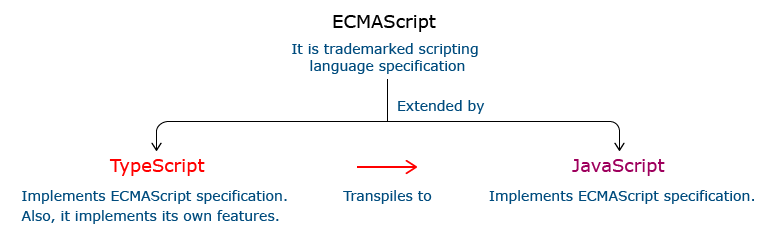
The solution can be to choose a language which is rich in features and the code can be converted to JavaScript for browsers. This process of converting code written in one language into another language is generally called Transpilation.

TypeScript is one such language whose code can be transpiled to JavaScript. This conversion is required because browser cannot understand TypeScript code.

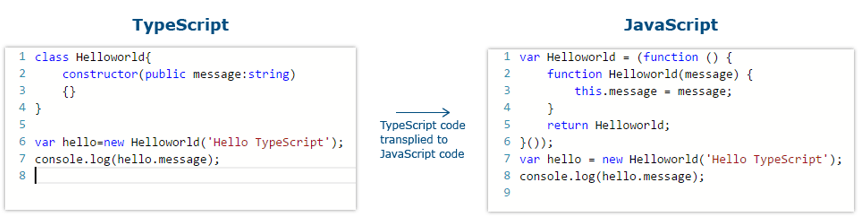
**What is TypeScript**

TypeScript is a typed superset of JavaScript that transpiles to JavaScript.

* TypeScript makes the development of JavaScript nearer to a more traditional object oriented experience.
* TypeScript is based on [ECMAScript](http://www.ecma-international.org/) 7 proposals.
* Apart from the EcmaScript specification, TypeScript has its own features as well.
* Any valid JavaScript is TypeScript.



**Relationship Between TypeScript and JavaScript**



**I**n the code given above, the TypeScript class HelloWorld is converted to a self invoking function in JavaScript when transpiled.

**Features Of TypeScript**

**Static Typing**: It adds static typing to JavaScript, due to which the readability of the code improves and also helps in finding more early compilation errors than the runtime errors.

**Modules support:**TypeScript provides an option to create modules so that we can modularize the code for easy maintenance. Modules also help in making the application scalable.

**Object Oriented Programming:**TypeScript supports object oriented programming features such as classes, encapsulation, interface, inheritance and so on which help in creating highly structured and reusable code.

**Open Source:**TypeScript is open source. The source code of TypeScript can be downloaded from github.

**Cross Platform: It works across platforms.**

**Tooling Support:**TypeScript works extremely well with Sublime Text, Eclipse, and almost all major IDEs as compared to JavaScript

**3.4.2 VISUAL STUDIO CODE IDE**

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open source and released under the permissive MIT License. The compiled binaries are freeware and free for private or commercial use.

Visual Studio Code is based on Electron, a framework which is used to deploy Node.js applications for the desktop running on the Blink layout engine. Although it uses the Electron framework, the software does not use Atom and instead employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

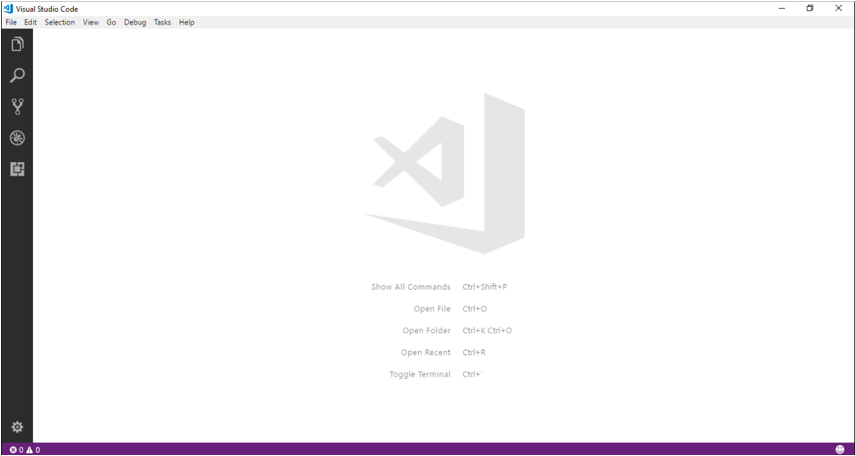
Typescript code can be written using several IDEs such as - Eclipse IDE, NetBeans IDE, Visual Studio Code IDE etc. We are prefer using Visual Studio Code IDE as it is a lightweight cross-platform editor which comes with built-in TypeScript support.

Typescript code is written in Visual Studio Code IDE in the following way:

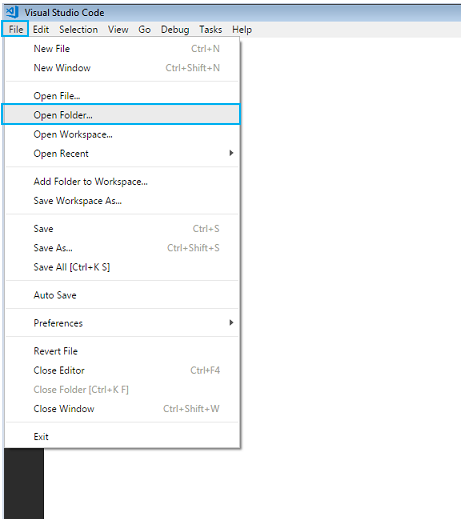
**Step 1:**Create a folder on your desktop. This folder will be used as a work-space for Visual Studio Code. Let us name it **TypeScript\_WorkSpace.**

**Step 2:** Create a folder and name it as **Demo**, inside the previously created folder. This folder will be used as a project folder.

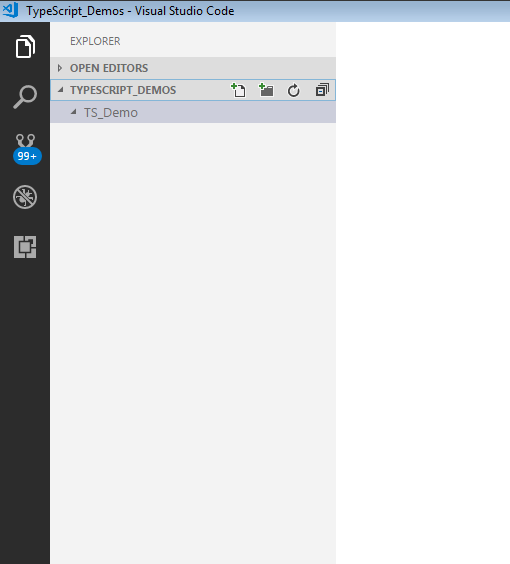
**Step 3:**Launch Visual Studio Code IDE. Once launched, close the welcome page. We should get a screen similar to the one shown below.



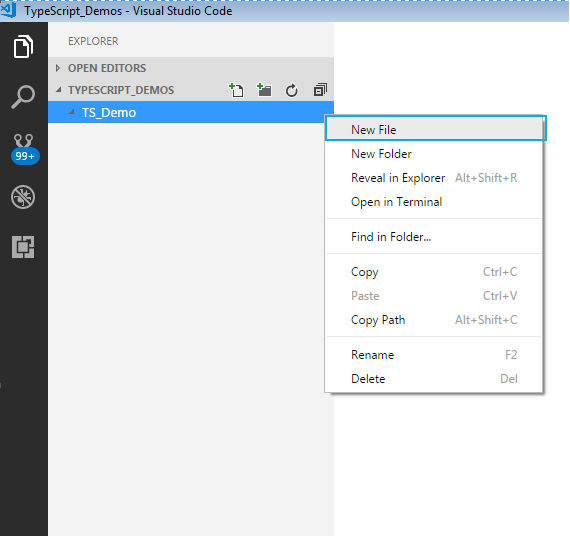
**Step 4:**Next step is to select a workspace. From the **File** menu, select **OpenFolder,** browse for the TypeScript\_Demos folder and select it.

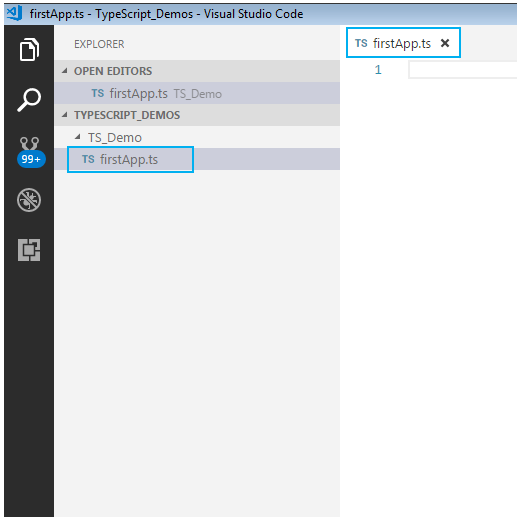


**Step 5:**The IDE will restart and the folder will be selected as work-space. We will also have the **TS\_Demo** project folder as shown below.



**Step 6:** Let us create a typescript file. Create a file named **firstApp.ts** in the project folder.





To start with the first application in TypeScript, in already created firstApp.ts file, give a console.log

statement and save it.

Here is how to do this:

**Step 7:** We shall print a welcome message when we run the above file. To do so, let us code as shown

**console.log("Hello! Welcome to TypeScript");**

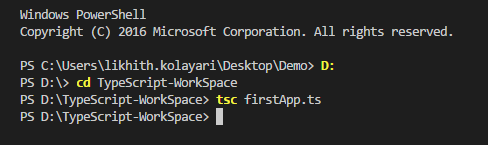
**Step 8:**

To execute the code, we will have to navigate to the project folder. For this, we will use windows command prompt through VSCode. Open the command prompt by following step:

* Navigate to View tab in VSCode.
* Select Integrated Terminal.

Alternatively, you can use the shortcut, **ctrl+`**, to open the command prompt. We will get the command prompt as shown below.

From the command prompt navigate to the folder in which the ts file resides and **transpile**the **.ts** file using the **tsc**command as follows:



**Step 9:** Run the transpiled **.js** file using **node** command. On execution of the code, the output is displayed on the console as shown.

