

A PROJECT REPORT
on
Trainer Pool Management
submitted for the partial fulfilment of the of the requirement
for the award of the degree of
Bachelor of Technology
in
Computer Science and Engineering
by
Sandeep K. Mohanty (1501021118)
Under the Supervision of
Shilpa Mahajani
Project Manager, CDC, Pune



DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved
by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

2018-2019



DECLARATION

This is to certify that the Project entitled “*Trainer Pool Management*” in partial fulfillment of the requirement for the award of the **Degree of Bachelor of Technology in Computer Science & Engineering**, submitted to **DIT University, Dehradun, Uttarakhand, India**, is an authentic record of my own work carried out during the period Jan 2019 to May 2019, under the supervision of Shilpa Mahajani (Supervisor and Project Mentor).

The matter embodied in this Project has not been submitted for the award of any other degree or diploma to any University/Institution.

Date:
Place: Pune

Sandeep Kumar Mohanty
1501021118

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:
Place: Pune

Shilpa Mahajani
Project Manager
Cognizant Tech. Solution



CERTIFICATE

This is to certify that the Project entitled “*Trainer Pool Management*” in partial fulfilment of the requirement for the award of the *Degree of Bachelor of Technology in Computer Science & Engineering*, submitted to *DIT University, Dehradun, Uttarakhand, India*, is an authentic record of bonafide project work carried out by Sandeep Kumar Mohanty Roll No. 1501021118 under supervision.

Shilpa Mahajani

Supervisor, Project Manager

Cognizant Tech Solutions, Pune

Date:

Place: Pune

ABBREVIATIONS

MVC	Model View Controller
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
DB	Database
SME	Subject Matter Expert
SQL	Structured Query Language
SDLC	Software Development Life Cycle
API	Application Programming Interface
JSON	Java Script Object Notation
JDBC	Java Database connectivity
XML	Extensive Markup Language

ACKNOWLEDGEMENT

For the successful completion of this project, I'd extend a sincere thanks to our project guide as well as trainer Shilpa Mahajani mam, who has been there with us while building the complete code from scratch. Without her guidance and teaching, it'd have been impossible of us to create the project.

I'm also grateful to our batch owners and people who've helped us out in every way possible, Manoj Kulkarni Sir, Mohit Kariya Sir, Gerard Thomas Sir and Komal Sanjay Pawar Ma'am. Without them, internship at such a huge company would have not been completed successfully. They ensured our smooth functioning and handled all the quirks and doubts while the sessions which helped us a lot in order to calmly finish the project.

I'd also like to thank my college for giving us such an amazing opportunity to work through the college semester in order to get a corporate experience and education. It is only because of his thought that we were able to gain such a training with a parallel balance of college academics.

ABSTRACT

This report is aimed at documenting the “Trainer Pool Management” that has been developed as a solution of a common interface between the allocators and the nominees. While starting with the project we went through some similar existing platforms that inspired the project to be developed in the first place, however it provided the motivation to do it even better. The system we have created provides a common platform for logging in / signing up as the predefined user types that are as mentioned, the allocators and the nominees and thereafter surfing their corresponding dashboards in order to complete their tasks. Their dashboards comprise of windows for their respective functions that is easy to navigate and control since the main objective of the management system has been kept as the same. In this project, the function of allocator is performed by the ‘Admin’ and nominees are the ‘Trainers’ and the ‘SMEs’. The trainers based on their skill sets and dates of availability, request for training a particular course and the SMEs nominate themselves to be a trainer with the same. Both of these applicants can update their profiles, skill sets or even dates once they have sent the request/ nomination. Also, the trainer can be allocated to multiple courses that fits in his available timeline and course schedule. The admin performs the task of managing a calendar which includes details of the program / course and the trainer allocated for the respective dates. His another function includes managing the database of trainers and SMEs along with upgrading the eligible and available SMEs to trainers as well as allocating the trainers to program / course based on his request and current requirement.

TABLE OF CONTENT

S.no.	TITLE	Page no.
A	ABBREVIATIONS	iv
B	ACKNOWLEDGEMENT	v
C	ABSTRACT	vi
D	LIST OF FIGURES	ix
1	CHAPTER 1 : INTRODUCTION	1
	1.1 Purpose	1
	1.2 Objective	1
	1.3 Motivation	2
	1.4 Definition and Overview	2
2	CHAPTER 2: AGILE METHODOLOGY	3
	2.1 What is Agile?	3
	2.2 What is Scrum?	3
	2.3. Scrum Rules	4
	2.4 Iterative vs Waterfall	5
	2.5 Agile Model – Pros and Cons	6
	2.6 Burn-down Chart	7
	2.7 User Stories	10
3	CHAPTER 4 : DETAILS OF TECHNOLOGIES USED	11
	3.1 MVC (Model-View-Controller)	11
	3.2 HTML	11

	3.3 CSS	12
	3.4 JavaScript	12
	3.5 Bootstrap	13
	3.6 Java Servlet	13
	3.7 JDBC	13
	3.8 MySQL	13
4	SYSTEM REQUIREMENTS	14
	4.1 Hardware Requirements	14
	4.2 Software Requirements	14
	4.3 Functional Requirements	14
	4.4 Non Functional Requirements	15
5	CHAPTER 5 : DESIGN	16
	5.1 Admin Flow Chart	16
	5.2 Trainer Flow Chart	17
	5.3 SME Flow Chart	18
	5.4 E-R Diagram	19
6	CHAPTER 6 : SNAPSHOTS	20
7	CHAPTER 7: CONCLUSION & FUTURE WORK	24
8	CHAPTER 8 : REFERENCE	25

List of Figures

Figure No.	Figure Name	Page No.
Fig 2.4.1	Phases of Iterative Waterfall Model	5
Fig 2.4.2	Iterative vs. Waterfall	6
Fig 2.6.1	Working of Iterations	7
Fig 2.6.2	Ideal Burn Down Graph	8
Fig 2.6.3	Realistic Burn Down Graph	9
Fig 2.6.4	Explanation	9
Fig 2.7.1	Iteration Level	10
Fig 5.1.1	Admin Flow Chart	16
Fig 5.2.1	Trainer Flow Chart	17
Fig 5.3.1	SME Flow Chart	18
Fig 5.4.1	E-R Diagram	19
Fig 6.1	Landing Page	20
Fig 6.2	Admin Login	20
Fig 6.3	Admin Dashboard	21
Fig 6.4	Trainer Allocation	21
Fig 6.5	Trainer Calendar	22
Fig 6.6	SME Dashboard	22
Fig 6.7	SME Create Nomination	23
Fig 6.8	Trainer Dashboard	23

CHAPTER – 1

INTRODUCTION

1.1 Purpose

This document is aimed at:

- Providing the necessary inputs to the detailed requirements gathering phase and further on for the SDLC processes.

The purpose of this document is to systematically capture requirements for the project and the system to be developed. Functional requirements are captured in this document. Trainer Pool management allows the nominees to register themselves by login or through signing up. Trainer can define their date of availability for the particular course and SME can nominate themselves as trainer for taking any course with the concern of the Admin that is only admin can allocate the course and define the period for the course.

All the details of the trainer, SME, and Admin are stored in the database.

1.2 Objective

Administrator

Below are the objectives of Administrator Module:

- User registration & credential authentication
- Allocate trainer with particular program
- Generate confirmation of trainer for allocated program
- Generate Calendar list of allocated Trainers

Trainer

Below are the objectives of Trainer Module:

- User registration & credential authentication
- Procurement of trainer information with skill group and availability
- Allocated program list for particular Trainer

SME

Below are the objectives of SME Module:

- User registration & credential authentication
- Procurement of trainer skill information and availability
- Allocated program list for particular SME

1.3 Motivation

This project was chosen with idea of automation of allocation of trainers and SME for the given course so that the nominee can book the knowing course easily and efficiently unlike those methods of going offline to institutes and paper advertisement. The proposed system should provide current details about the course, trainer available date, SME's on a web portal.

Any trainer and SME can register themselves. Trainer can provide the courses he/she can choose for teaching and their availability period. SME can nominate itself to trainer and then perform the same. Admin will look after the availability and then assign the trainer for the course.

1.4 Definition and overview

The 'Trainer Pool Management' is a web based application. Features of this web-application are as follows:

- 1.4.1 Admin/ Trainer/ SME login.
- 1.4.2 Trainer / SME signup.
- 1.4.3 Admin can allocate the trainer.
- 1.4.4 Trainer can create request and confirm it.
- 1.4.5 SME can nominate themselves to trainer.
- 1.4.6 Multi-user application & easy to use.

CHAPTER-2

AGILE METHODOLOGY

2.1 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.

2.2 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 is distinguished from other agile processes by specific concepts and practices, divided into the three categories of Roles, Artefacts, 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” 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

2.3 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.

2.4 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.

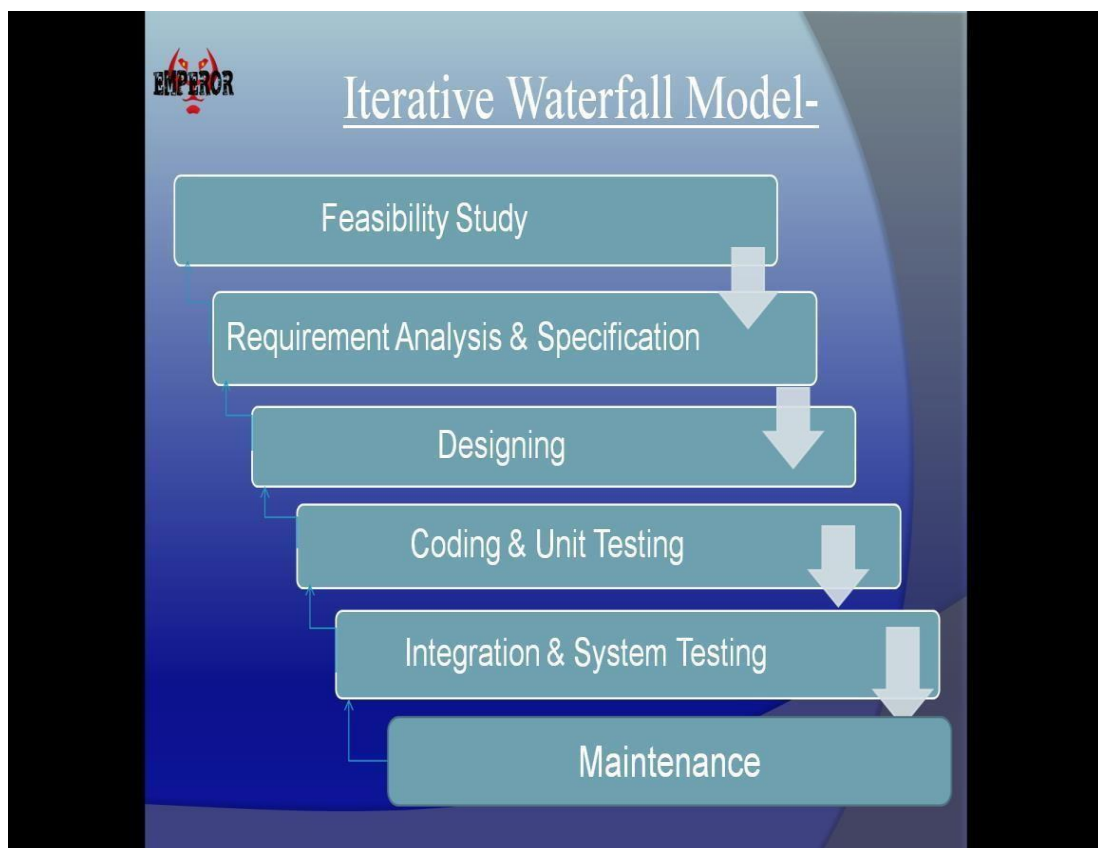


Fig. 2.4.1. Phases of Iterative Waterfall Model

This iterative approach supports a *product* rather than a *project* mind-set. 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 start-up concept.

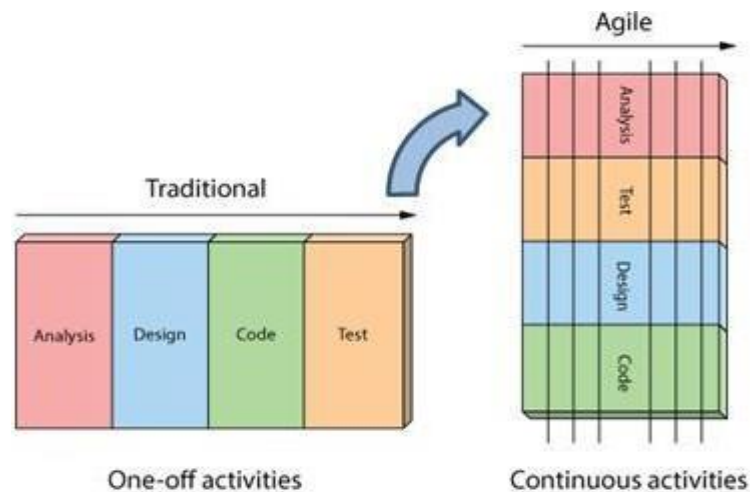


Fig. 2.4.2. Iterative vs. waterfall

2.5 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 –

- 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.

2.6 Burn-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.

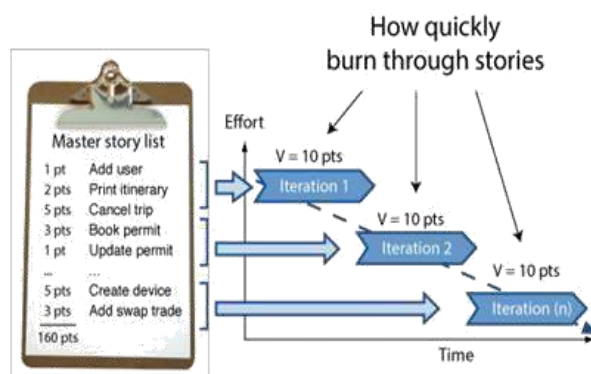


Figure 2.6.1 Working of Iterations

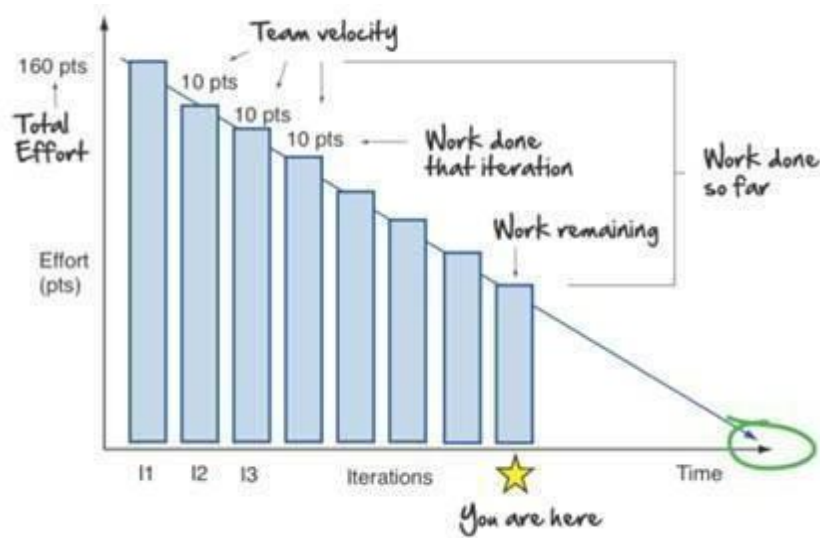


Fig.2.6.2. Ideal Burn Down Graph

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

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

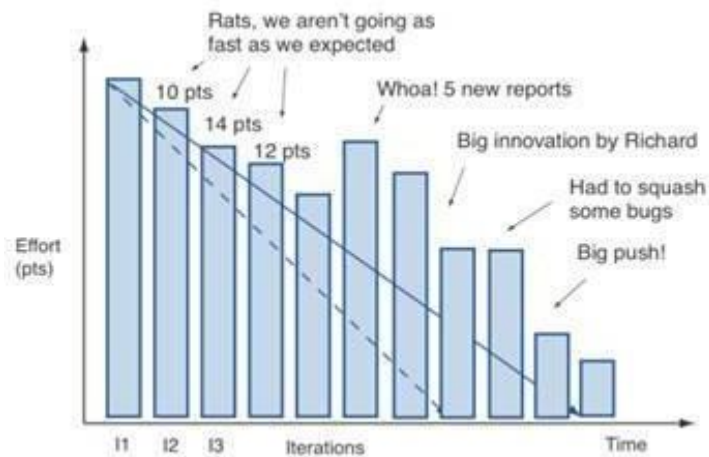


Fig. 2.6.3. Realistic Burn Down Graph

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 1/2 way done the project.

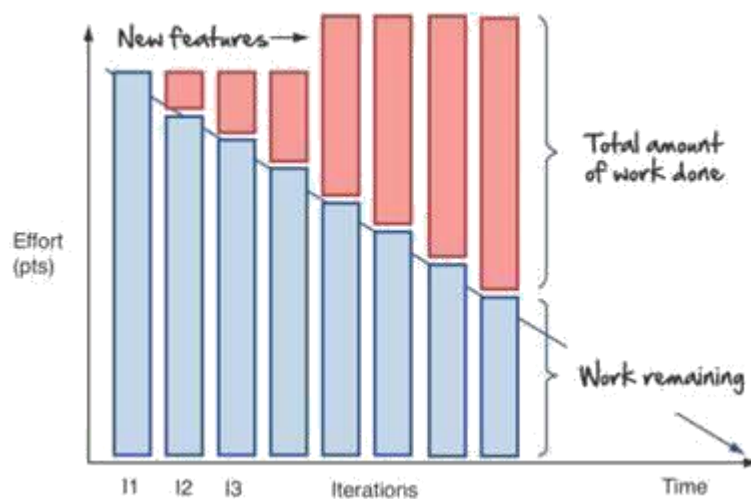


Fig.2.6.4. Explanation

2.7 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.2.7.1. Iteration level

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

Typically, no more than a couple of 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. User stories are one of the primary development artefacts for Scrum and Extreme Programming (XP) project teams. A user story is a very high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimate of the effort to implement it.

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 mock-ups, anything that helps) and break the functionality down into simple easy to understand words and phrases our customers understand. User stories!

CHAPTER-3

DETAILS OF TECHNOLOGIES USED

3.1 MVC (Model –View-Controller)

Model–view–controller is commonly used for developing software that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

Traditionally used for desktop graphical user interfaces (GUIs), this architecture has become popular for designing web applications and even mobile, desktop and other clients.

Popular programming languages like Java, C#, Ruby, PHP and others have popular MVC frameworks that are currently being used in web application development straight out of the box.

3.2 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 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.

3.3 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, colours, 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.

3.4 JavaScript

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.

3.5 Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

3.6 Java Servlet

A Java servlet processes or stores a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any client-server protocol, but they are most often used with the HTTP.

Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML and more commonly, JSON. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL mapping.

3.7 JDBC

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases.

3.8 MySQL

MySQL is an open source relational database management system (RDBMS).

"SQL", is abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).

CHAPTER-4

SYSTEM REQUIREMENTS

4.1 Hardware Requirements:

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

4.2 Software Requirements:

Operating System	:	Windows XP, Windows 8/8.1, Windows 10.
RAM	:	Minimum 256 MB.
IDE	:	Eclipse, Net Beans.
Database Server	:	SQL Server 2012, MySQL.
Web Server	:	Tomcat version 7.0 and above.

4.3 Functional Requirement

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

This provides the admin the facility to perform specific task assigned to such as he should be able to view all the trainers and SME's. He should be able to view their availability duration.

It should allow the Admin/trainer/SMEs to login from any system and he/she can access the shared path.

4.4 Non-functional Requirement

- AVAILABILITY: This application is available online.
- RELIABILITY : This application works same according to documentation.
- FLEXIBILITY : This application works on and above every platform.

CHAPTER - 5

DESIGN

5.1 Admin Flow Chart

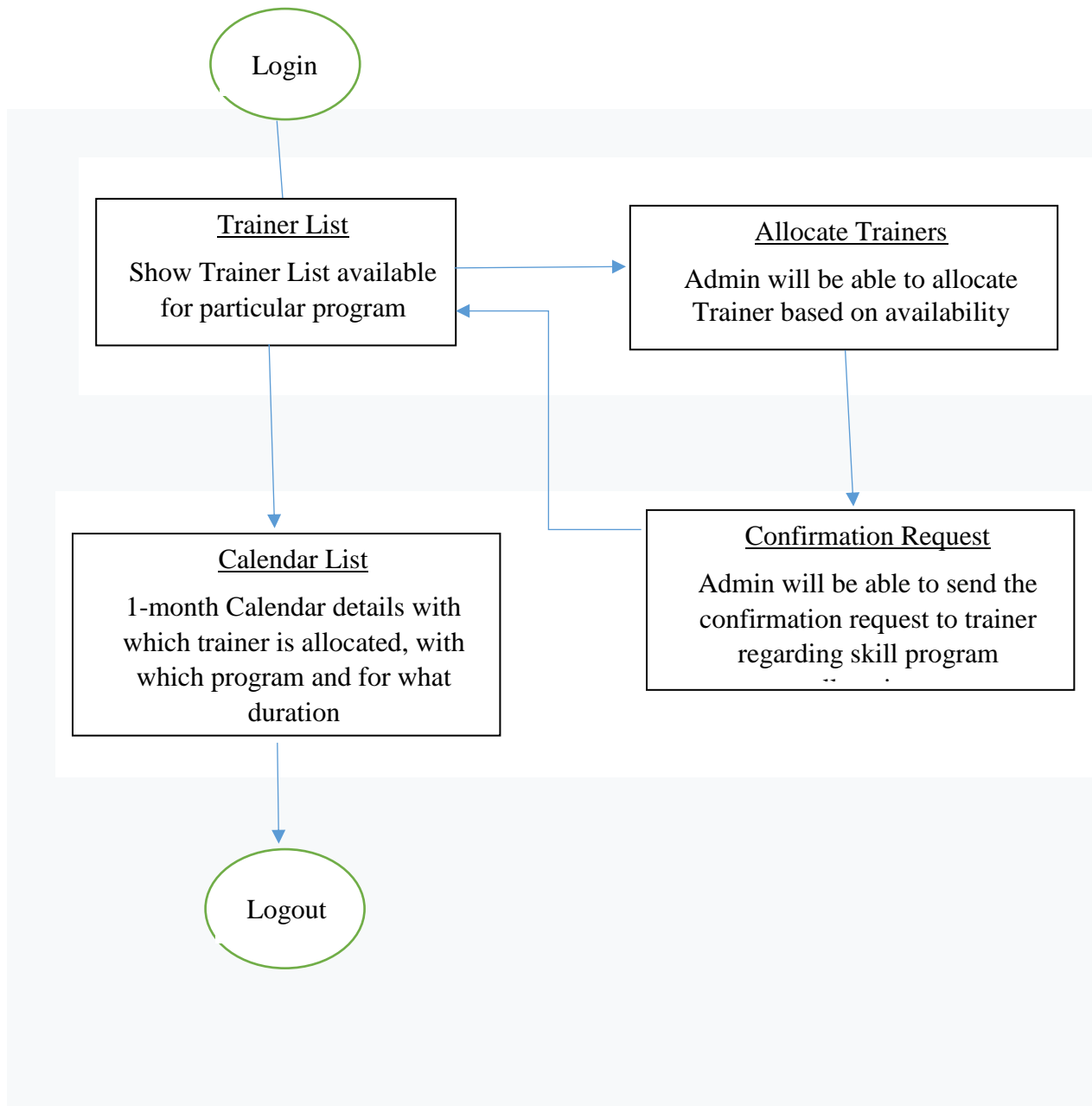


Fig 5.1.1 Admin Flow Chart

5.2 Trainer Flow Chart

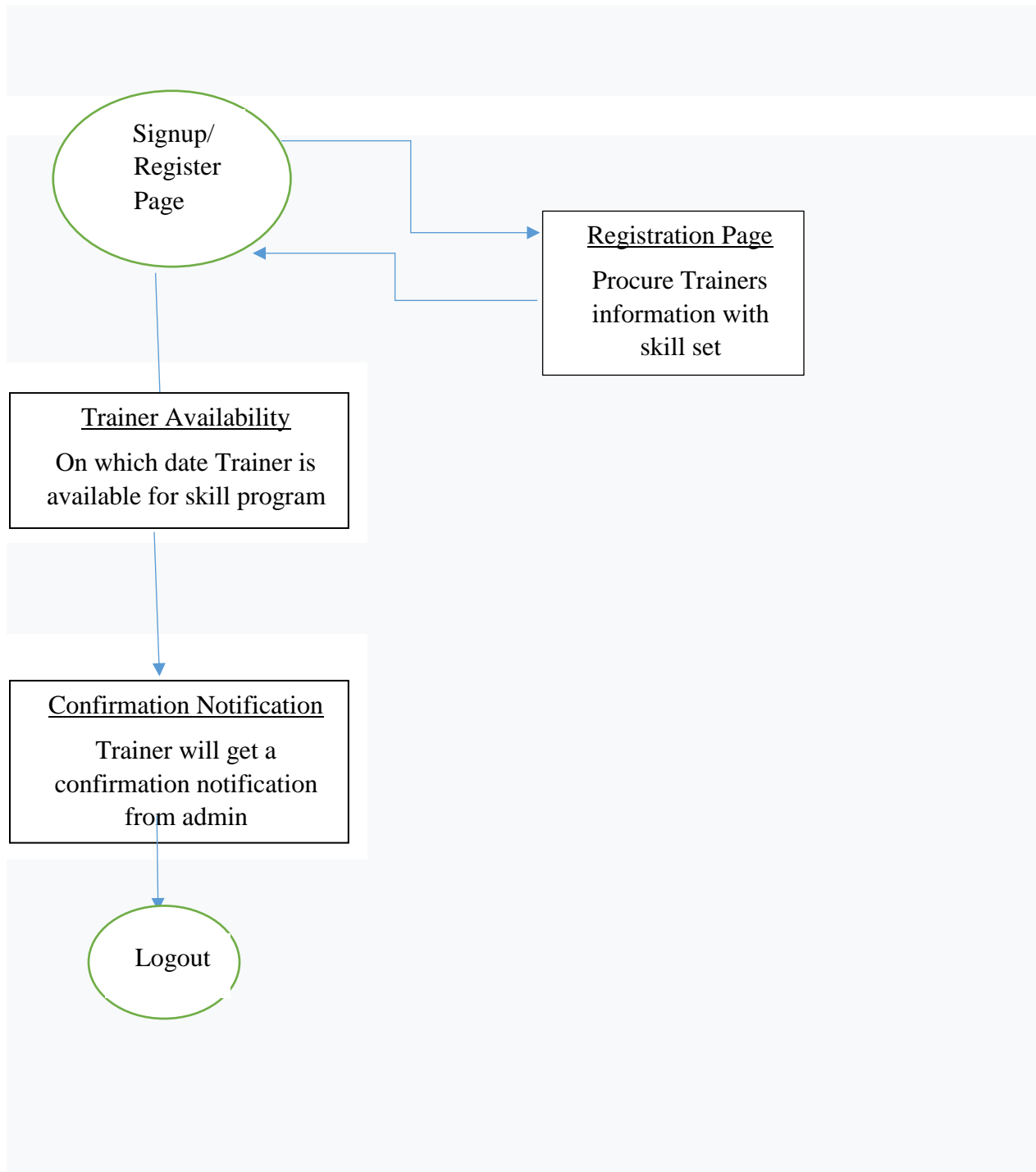


Fig 5.2.1 Trainer Flow Chart

5.3 SME Flow Chart

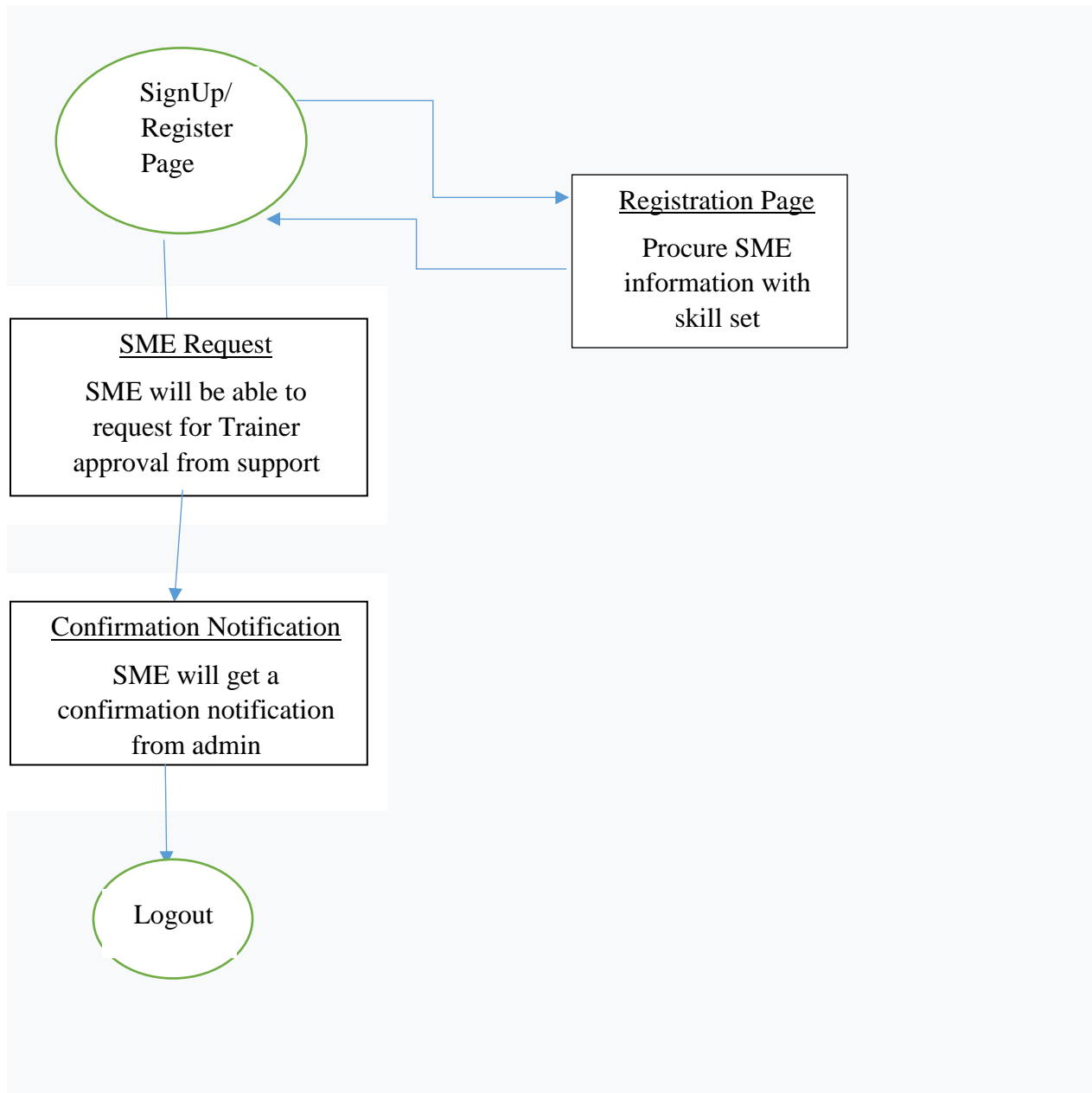


Fig 5.3.1 SME Flow Chart

5.4 E-R Diagram

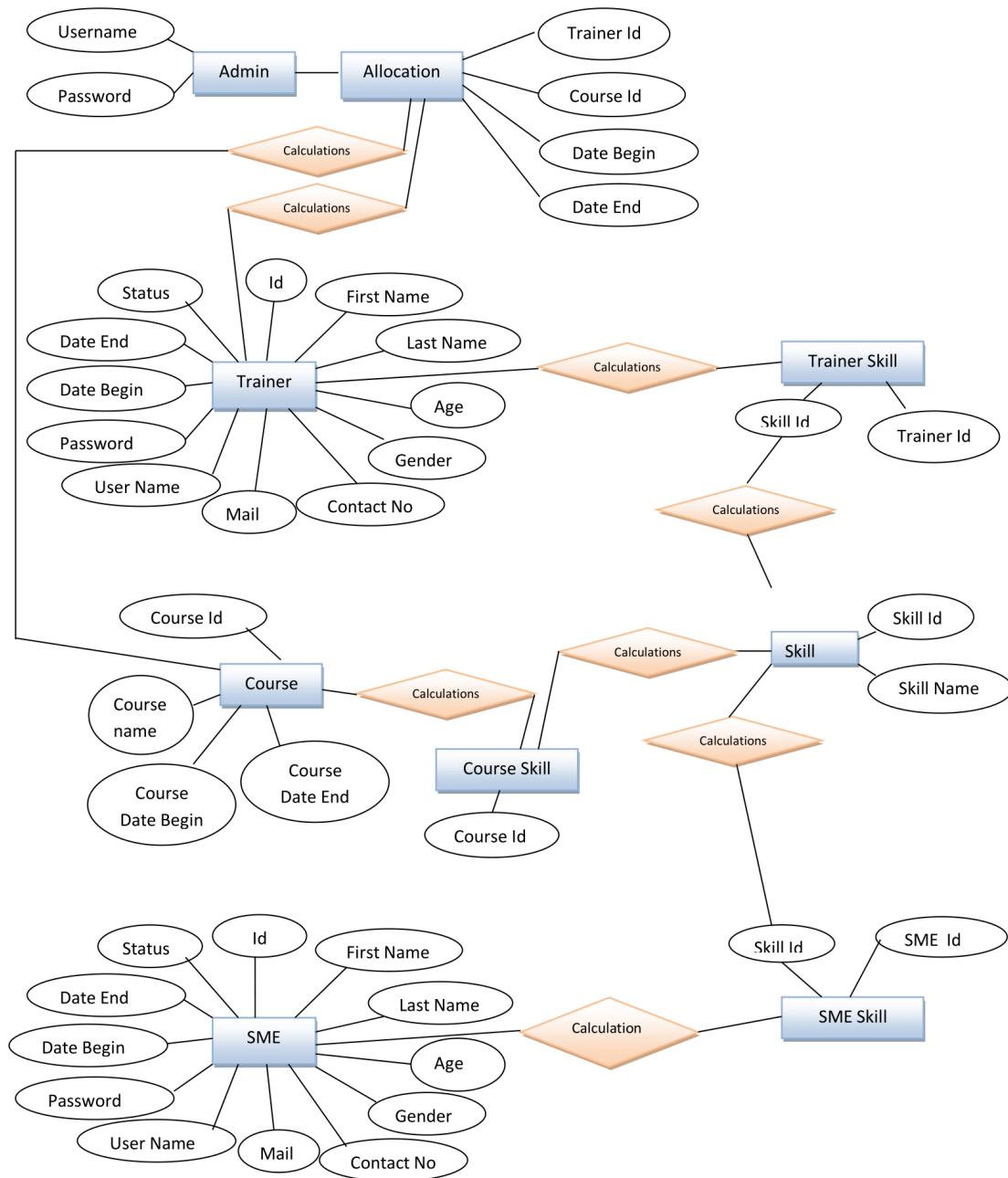


Fig 5.4.1 E-R Diagram

CHAPTER – 6

SNAPSHOTS

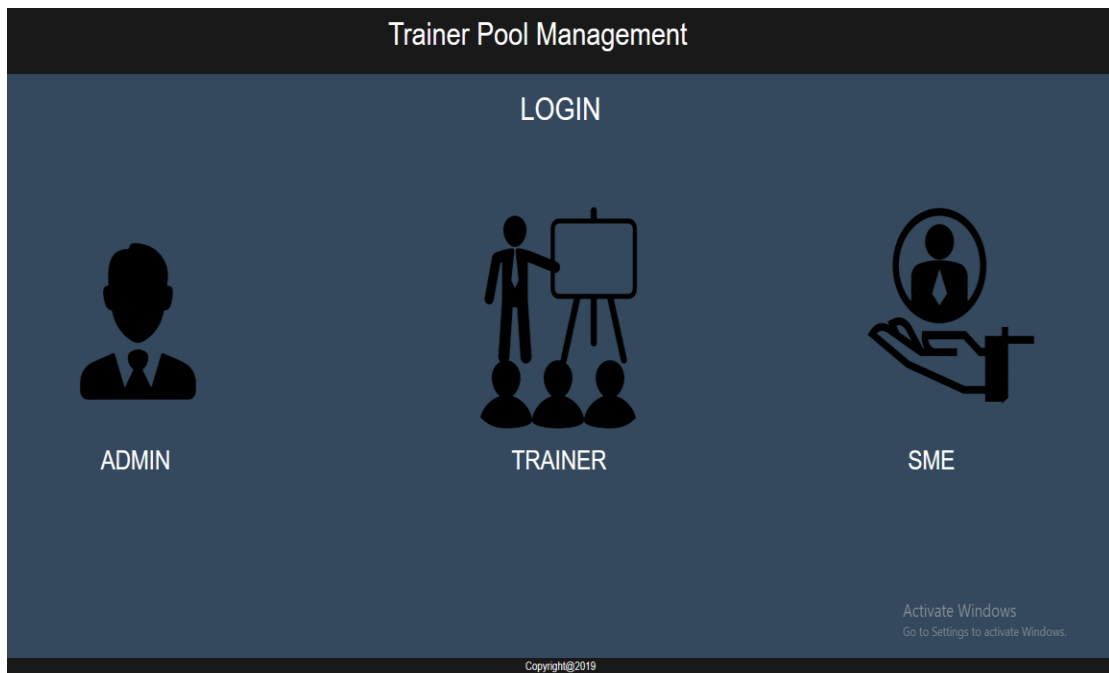


Fig 6.1 Landing Page

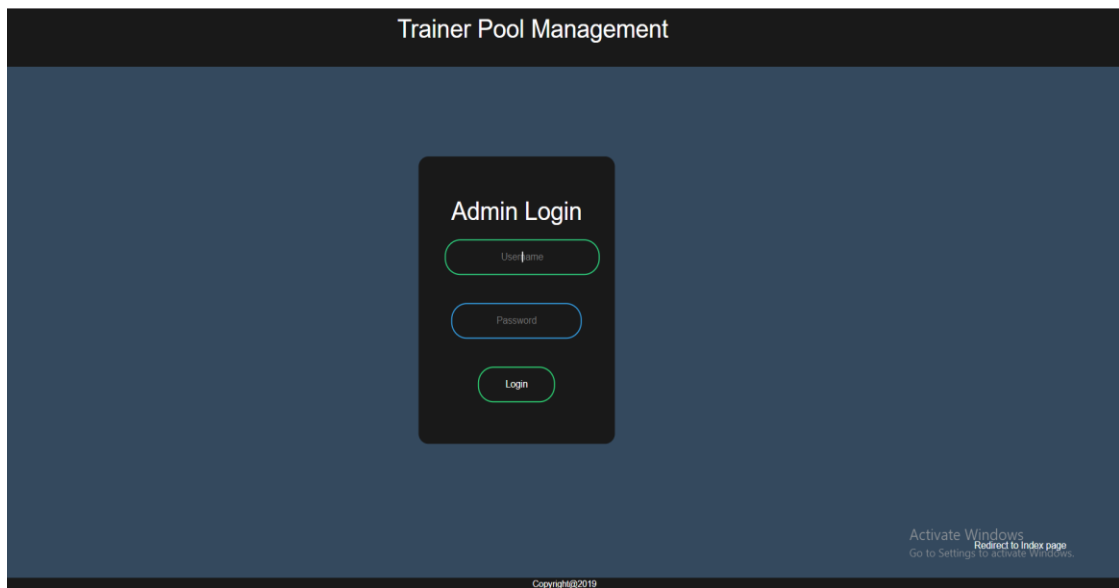


Fig 6.2 Admin Login Page

CHAPTER - 7

CONCLUSION AND FUTURE WORK

The trainer pool management system guided us through the proper architecture of a management system, making us learn the structured levels of development. Being new to the software, a little problem was faced while writing the complete code from scratch however learning the new technologies in order to build the project has leveled up the making process of management system. As a limitation of project, there could be a few more functionalities added to enhance the system and similarly a better architecture can be used to make it smoother. But whatever is made, it has surely cleared our basics and taught us a lot. Concluding to it, we look forward to enhance the features of the system as well as expanding the possible opportunities that come its way.

CHAPTER – 8

REFERENCES

- w3schools.com
- docs.oracle.com
- stackoverflow.com
- css-tricks.com
- www.techonthenet.com
- www.tutorialspoint.com
- beginnersbook.com

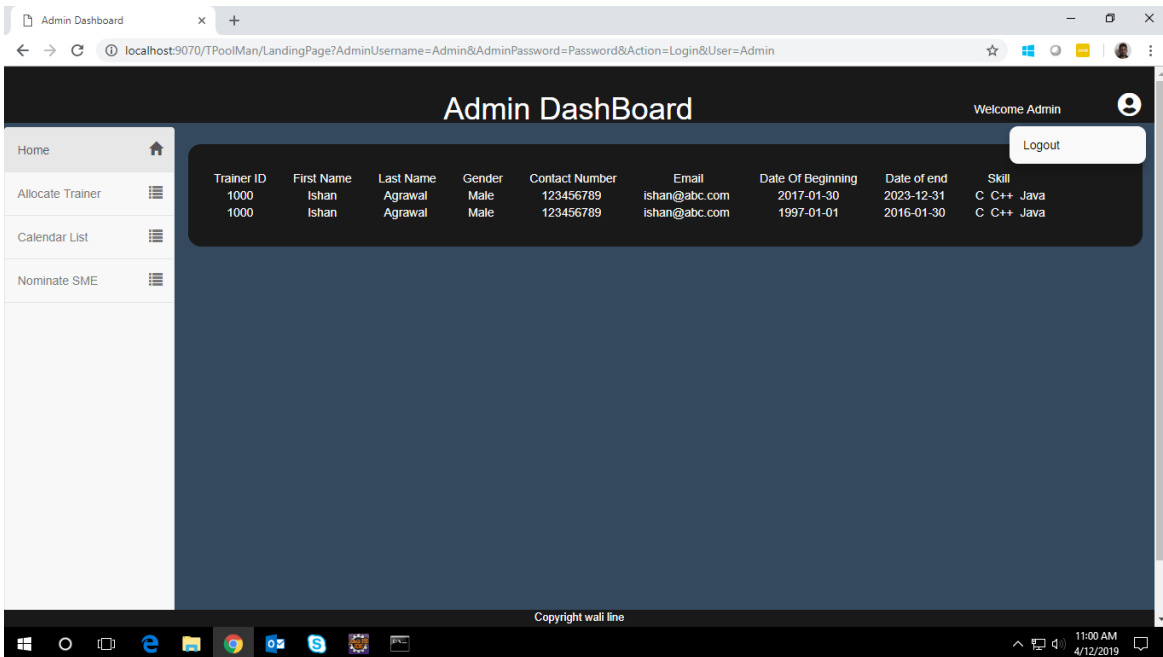


Fig 6.3 Admin Dashboard

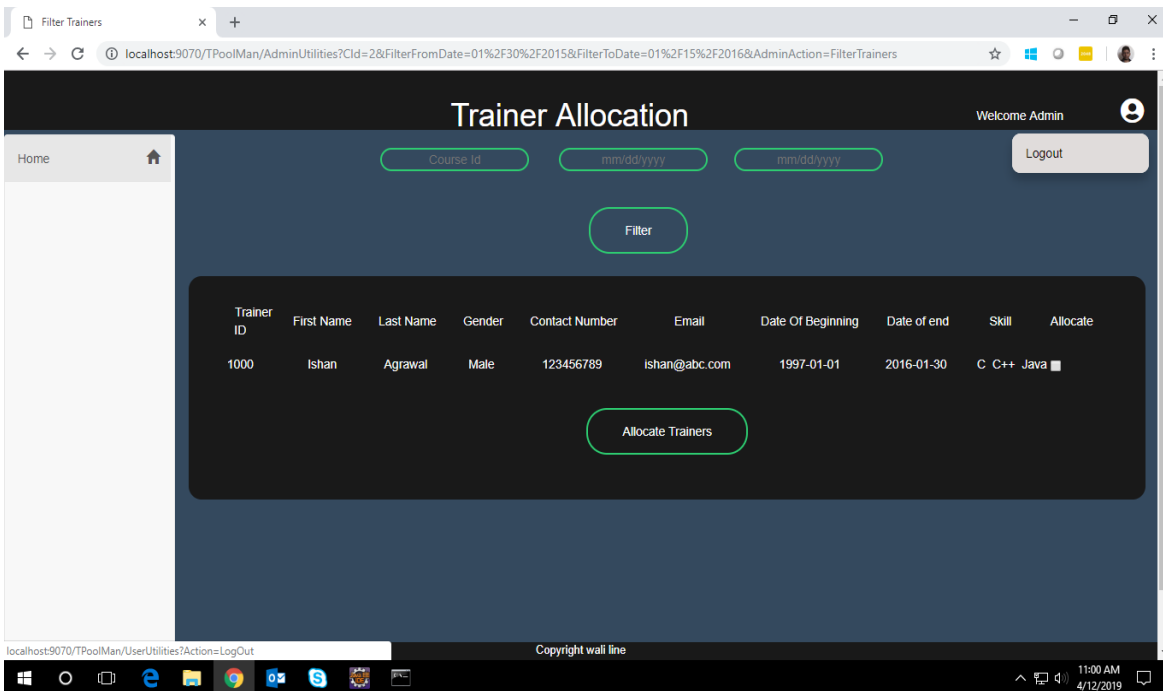


Fig 6.4 Trainer Allocation

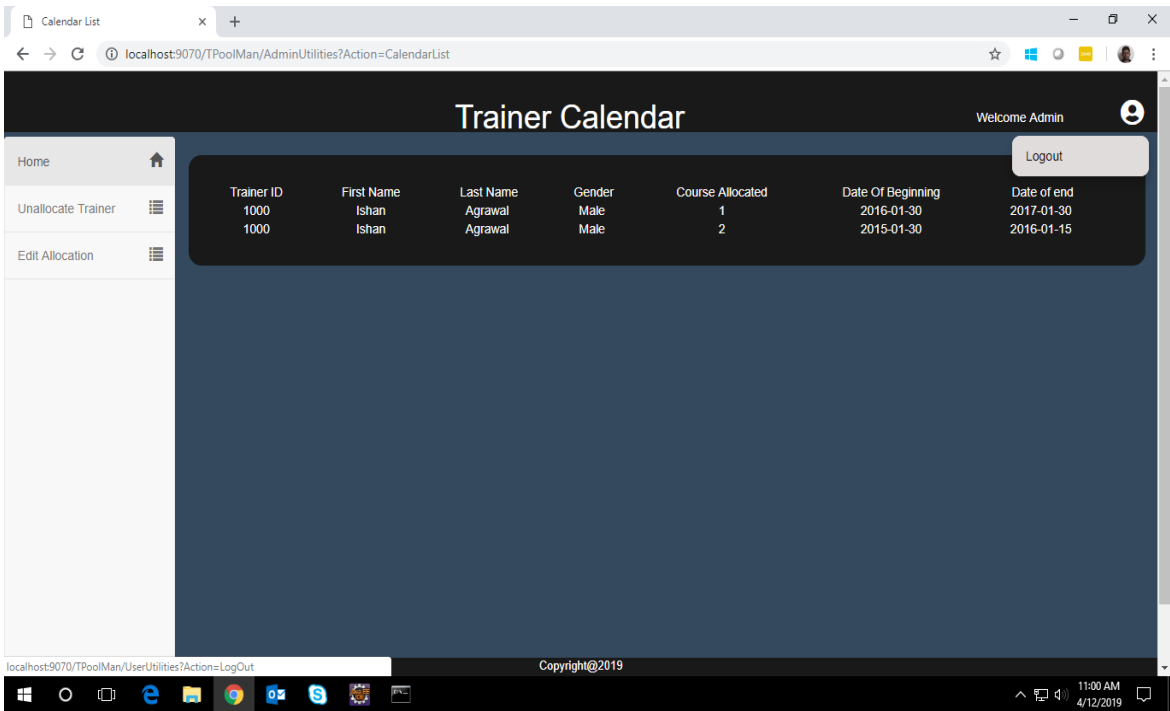


Fig 6.5 Trainer Calendar

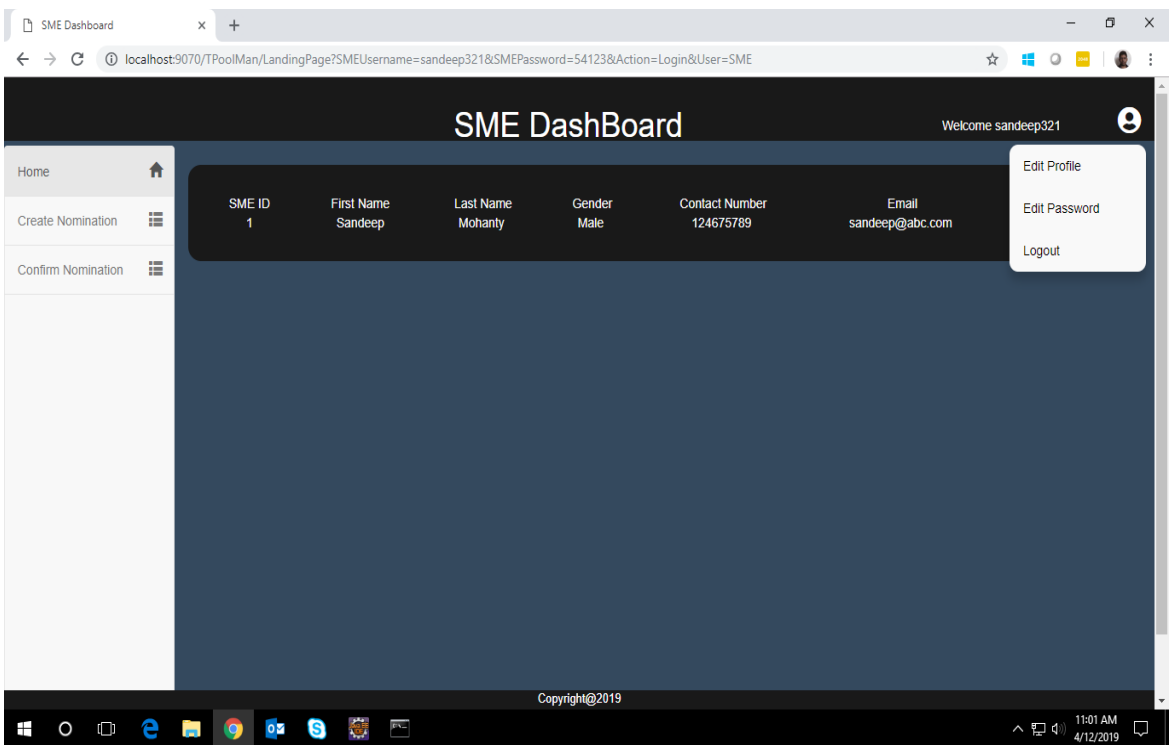


Fig 6.6 SME Dashboard

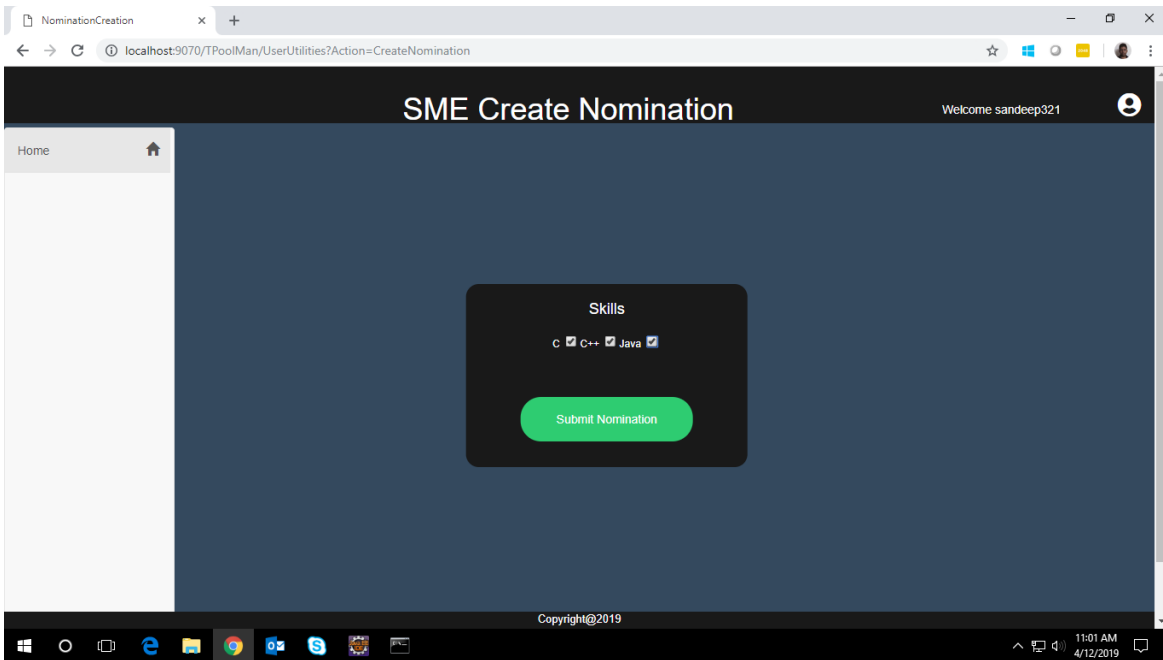


Fig 6.7 SME Create Nomination

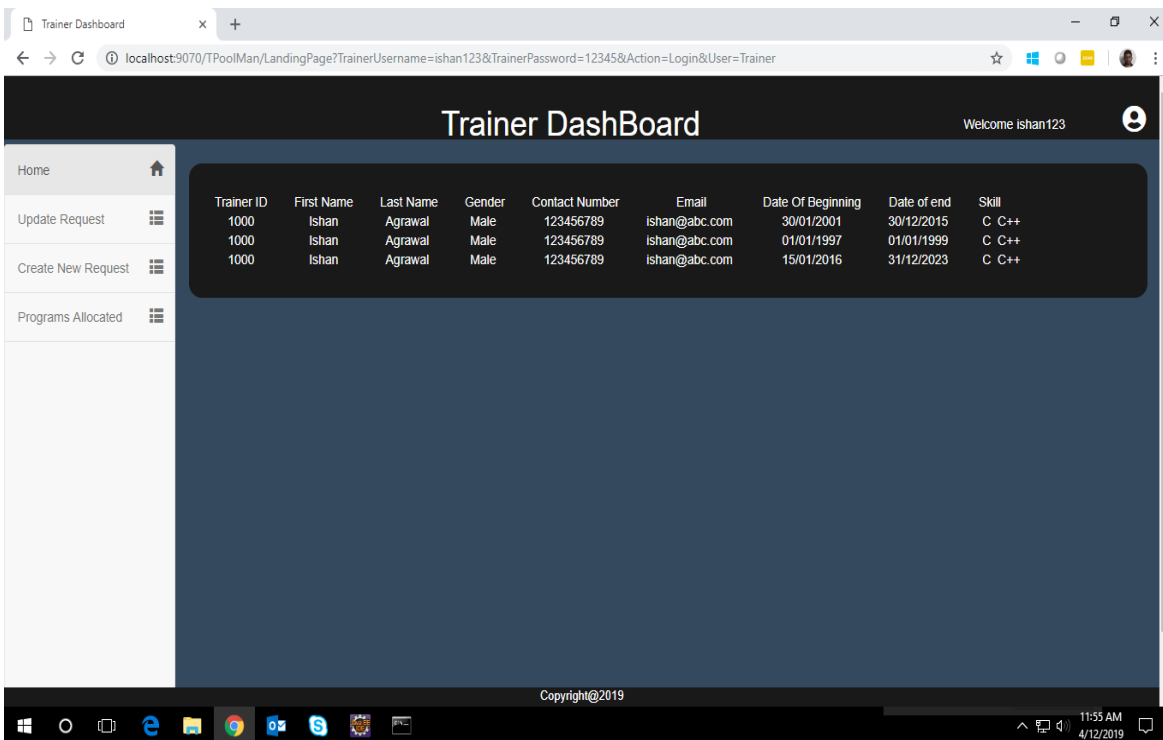


Fig 6.8 Trainer Dashboard

