**Work on Angular**

**eClerx Services Limited**

**A training report**

Submitted in partial fulfilment of the requirements for the award of degree of

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted to**

**LOVELY PROFESSIONAL UNIVERSITY**

# PHAGWARA, PUNJAB



**From 08/06/19 to 12/14/19**

**Name of student: Fairy Sharma Name of Supervisor: Shashi Ojha**

**Registration Number: 11607222 Designation: Senior Process Manger**

**Signature of the student:**

**Student Declaration**

# To whom so ever it may concern

I, **Fairy Sharma, 11607222,** hereby declare that the work done by me on “**Angular**” from **August, 2019** to **December, 2019**, under the supervision of **Shashi Ojha**, **Senior Process Manger, eclerx Services Limited,** and **Poonam Thakur**, **Assistant Professor,** Lovely professional University, Phagwara, Punjab, is a record of original work for the partial fulfillment of the requirements for the award of the degree, **Bachelor of Computer Science Engineering.**

Name of the Student (Registration Number): - Fairy Sharma (11607222)

Signature of the student

Dated: - 14/December/2019

**Declaration by the supervisors**

# To whom so ever it may concern

This is to certify that **Fairy Sharma**, **11607222** from Lovely Professional

University, Phagwara, Punjab, has worked as a trainee in **eclerx Services Limited** on “**Angular**” under my supervision from **August, 2019** to **December, 2019**. It is further stated that the work carried out by the student is a record of original work to the best of my knowledge for the partial fulfilment of the requirements for the award of the degree, degree name.

**External Supervisor Internal Supervisor**

|  |  |
| --- | --- |
| Name: - Shashi Ojha | Name: - Deepak Prasher |
| Designation: - Senior Process Manager | Designation: - HOD |
| Signature | Signature |
| Dated: 14/Dec/2019 | Dated: 14/Dec/2019 |
|  |  |

**Acknowledgement**

I would like to express my gratitude to my manager, Mr Shashi Ojha (Senior Process Manger in eClerx Services Limited), who has been guiding me through the period of the internship.

I would like to thank u my Internal supervisor and my mentor in my college Mr Deepak Prashar (Head of Department at Lovely Professional University) and Mrs Poonam Thakur (Assistant Professor at Lovely Professional University). I would also like to thank LPU for paving the path for me.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S no.** | **TITLE** | **PAGE** |
| **1** | **Chapter 1 INTRODUCTION OF THE COMPANY** | 6 |
| **2** | **Chapter 2 INTRODUCTION TO THE PROJECT UNDERTAKEN** | 13 |
| **3** | **Chapter 3 BRIEF DESCRIPTION OF THE WORK DONE** | 28 |
| **4** | **Chapter 4 CARS MANAGEMENT PROJECT** | 38 |
| **5** | **Chapter 5 CONCLUSION** | 44 |
| **6** | **REFERENCES** | 45 |



**CHAPTER- 1**

# INTRODUCTION OF THE COMPANY

* **Company’s Vision and Mission**: -
* **Creating enduring value for those most in need, for our communities and our world: -** No matter our achievements or number of clients, philanthropy is fundamental to who we are. eClerx has a strong vision of what it means to be a responsible company in today’s society. We place great importance on the role our firm and employees can play in the communities in which we live and work. As we grow and expand our capabilities and innovations in the marketplace, we remain cognizant of our responsibility as change agents in our communities. That’s why we are committed to facilitating, funding and supporting social and economic transformations across various underserved communities.

At eClerx, we believe that when you educate a child you can transform the world. Talent and resources should go beyond optimizing businesses. They are part and parcel of expanding individual potential and helping them lead better, more productive lives. That’s why we’re committed to working toward a better future for the most disadvantaged and vulnerable populations in our world.

* **eClerx Cares: -** Children rights and education are central to our Corporate Social Responsibility mission. The eClerx family is active in helping to measurably and demonstrably improve the lives of underprivileged children. eClerx Cares, our India based CSR Committee operates under the guidance of eClerx Board of Directors to coordinate and manage our Corporate Social Responsibility initiatives.

To maximize our impact, we actively partner with NGOs and education stakeholders to find and advance effective solutions that address today’s most pressing development and educational challenges. Our partners work to secure the well-being of disadvantaged Indian youth by supporting education and child welfare initiatives throughout India. This includes vocational education and training for children living in city slums, rural villages and tribal areas as well as those who are unemployed or minimally employed. The eClerx Cares Committee approves and monitors projects we fund through our NGO partners. The eClerx Cares Council is responsible for coordinating and advancing our employee engagement initiatives, for example, eClerx employees have volunteered ad-funded efforts that have brought substantial impact to more than 30,000 lives.

* **Our Work with NGO Partners: -** As part of our social responsibility goals

eClerx is proud to partner with the following non-governmental organizations.

* eClerx provides critical business operations services to over fifty global Fortune 500 clients, including some of the world’s leading companies across financial services, cable & telecom, retail, fashion, media & entertainment, manufacturing, travel & leisure, software and high-tech. Incorporated in 2000, eClerx is one of India’s leading process management and data analytics companies and is today traded on both the Bombay and National Stock Exchanges of India. eClerx employs 9,000 people across its global sites in the US, UK, India, Italy, Germany, Singapore, and Thailand.
* **Deep Domain: -** We have in-depth knowledge in creating multi-client experiences, as well as industry and product expertise across the areas we

specialize in.

* **Automate & Re-engineer: -** We leverage our technology, analytics and industry expertise to reengineer, automate and improve functions with a focus on business outcomes and cost reduction.
* **Complex Functions: -** We specialize in transitioning, managing and transforming complex business-facing functions for many of the world’s largest organizations

* **Origin and growth of company**
* **Origin: -** The company founded in March, 2000. PD Mundra and Anjan Malik are the founder of this organisation. The Company headquarter is in Mumbai, India. There are number of employees under many process around 10000 employees are there in our company.

Our Company was originally incorporated on March 24, 2000 as eClerx Services Private Limited under the Companies Act,

1956. Our Company was converted to a public limited company and the name was changed to eClerx Services Limited, pursuant to resolutions of the shareholders passed in the AGM held on August 1, 2007. The Registrar of Companies, Mumbai issued a fresh certificate of incorporation consequent to the conversion on August 28, 2007.

eClerx is an Indian IT consulting and outsourcing multinational company based in Mumbai and Pune. eClerx is a public limited company whose shares are listed on the Bombay Stock Exchange and National Stock Exchange of India.

The company listed on the bourses on Monday, 31 December 2007. In April 2012, eClerx acquired Agilyst Inc to grow its footprint in the U.S. cable and telecom industry.

On 31 March 2015, eClerx acquired Creative Services Agency CLX Europe, for a total purchase consideration of up to EUR 25 million.

* **Growth of company: -** 
  + **Excellence** **-** Excellence is a journey - we achieve our goals by constantly improving, innovating and applying the highest standards.
  + **Integrity** **-** We are thoughtful, honest and empathetic in our interactions with clients, vendors, shareholders and each other.
  + **Client Centric** **-** Our clients' success is our goal – everything we do keeps their best interest at the forefront.
  + **People -** We care deeply about our people – we encourage learning, promote growth and celebrate diversity.
  + **Learning -** Our employees are a crucial part of our success – we strive to nurture and develop our employees with best-in-class learning opportunities
  + **Managing complexity while minimizing cost -** eClerx provides critical business operations services to over fifty global Fortune 500 clients, including some of the world’s leading companies across financial services, cable & telecom, retail, fashion, media &, manufacturing, travel & leisure, software and high-tech. Incorporated in 2000, eClerx is one of India’s leading process management and data analytics companies and is today traded on both the Bombay and National Stock Exchanges of India. eClerx employs 9,000+ people across its global sites in the US, UK, India, Italy, Germany, Singapore, Thailand
* **Various departments and their functions**
* **Digital –** eClerx Digital is the trusted partner of choice to the world’s largest global brands for creative production, eCommerce / web operations, and analytics & insights services. We improve profitability for their digital businesses. Our team of 3000+ full-time digital delivery employees at our five production hubs in Mumbai, Pune, Chandigarh, Verona and Phuket apply deep digital expertise to effectively support the most demanding global clients by employing a follow the sun delivery model. eClerx Digital’s innovative delivery model drives the “metrics that matter” for our clients: improved acquisition, conversion and retention and overall lifetime value of your customer 24x7x365**.**
  + **Marketing -** Maximize your marketing efforts to build awareness, generate leads, and increase sales.
  + **eCommerce -** Remove the friction from the user experience to reduce cart abandonment and increase average order value.
  + **Business Intelligence & Analytics -** Make sense of and use your data for personalization, consistently accurate forecasting, and to direct resources towards profitable activities.
  + **Operations & Finance -** Understand and optimize varied data and system processes for maximum return
  + **Creative -** Create stunning assets for use in print and digital
* **Financial Markets -** For financial organizations across the world, eClerx Markets, offers consulting, technological innovation, and process management expertise to uniquely solve operational challenges. With nearly two decades of industry experience complemented by the application of smart automation and robotics, our team of experts deliver holistic solutions across the trade life cycle, change management, data analytics, compliance, cash securities operations, document digitization and generation, and outreach
* **Trade Support** – Supporting post-executions activities across all instrument’s types, both cash and derivatives.
* **Client LifeCycle** – Gather, validate, digitize and distribute client related information, ensuring compliance with local regulations across multiple jurisdictions
* **Reference Data** – Organisation-wide data management solutions covering client, product, market and risk-related information
* **Asset Servicing** – Supporting management of all corporate actions, income collection, claim settlements and position reconciliations
* **Settlements and clearing** – A complete service covering position and cash reconciliations, static data management, collateral management and forecasting
* **Analytics** – Full spectrum of analytical services from business intelligence and digital analytics, to predictive modeling and forecasting.
* **Remediation Engagement / Project work** - Supporting change functions across all middle- and back-office functions, with expertise in Project Management, Project
* **Customer Operations -** eClerx improves customer experience and operational efficiencies by leveraging deep domain knowledge, advanced automation, and data analytics.

eClerx Customer Operations specializes in providing operational expertise and process excellence throughout the customer journey. We create solutions and services, utilizing a blend of technology and domain knowledge that support our clients’ evolving needs. Our suite of offerings enhances the customer experience by providing quality monitoring/insights, advanced analytics, automation, superior technical operations support and digital care services. We assist companies in developing, implementing and operating multichannel customer interaction capabilities – transforming everyday touchpoints into a superior customer experience**.**

* **Contact Quality Monitoring & Insight -** Customer Interaction Monitoring Mystery Shopping
* **Advanced Technical Operations -** Avoidable Truck Rolls Dispatch Services Reverse Logistics
* **Digital Care Services -** Advanced Chat Services Advanced Technical Support
* **Analytics & Automation** - Automation and AI Business Insights and Analytics Platforms Tools
* **Organization chart of the company: -** 





**CHAPTER- 2**

# INTRODUCTION TO THE PROJECT UNDERTAKEN

* **Objectives of the work undertaken**: -
* **Angular: -** AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications.
* It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. (This flexibility has led to the acronym MVW, which stands for "model-view-whatever" and may also encompass model–view–presenter and model–view–adapter.) In 2014, the original AngularJS team began working on the Angular web framework.
* Angular is a platform and framework for building client application in HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionalities as a set of TypeScript libraries that we import into our app.
* Components define views, which are sets of screen elements that angular can choose among and modify according to program logic and data. Components use services, which provide specific functionality not directly related to views. Service providers can be injected into the components as dependencies, this make the code modular, reusable and efficient.
* Components and services are simply classes, with decorators they provide metadata that tells angular how to use them.
* Angular apps are Modular and has its own modularity system NgModules. NgModules are containers for a cohesive block of code dedicated to an application domain, a workflow.
* They can import functionality that is exported from other NgModules, and export selected functionality for use by other NgModule. NgModules are the basic building blocks of angular applications, which provide a compilation context for components.
* Angular app has at least one module that is root module which enables bootstrapping, and has many more feature modules.
* NgModule is defined by a class decorated with @NgModule(). This decorator is a function that takes a single metadata object, whose properties describe the module.
* The objectives of this training are

* + Utilizing AngularJS formats adequately
  + Questioning and adjusting information in various databases and getting to be plainly gifted with the API
  + Quickly making perplexing structures
  + Understanding two-way (proportional) information authoritative
  + Presenting route usefulness in web applications
  + Overseeing conditions with Injection frameworks
  + Confining web applications to take into account worldwide groups of onlookers
  + Securing web applications from dangers and pernicious clients
  + Building different AngularJS orders
  + Understanding the compiler for building better and more propelled orders
  + Utilizing the testing system (Jasmine BDD) to test the web applications
  + Organizing the web application utilizing the vigorous index structure
  + Organizing, composing, and ultimately sending the application

* **Scope of the Work**

Ask any front-end developer today and they’ll tell you React is all the hotness. React can work on the web, your mobile device, your watch, and eventually your hair dryer,

fridge, and internet-connected T-shirt. While I like working with React just as much as the next front-end developer, I also have a pretty long history with Angular.

After picking it up in 2012, Angular was the first framework to give me a taste of what a JavaScript framework even is. Now that Angular 6 is out and all the chaos regarding the name change and framework change is behind us, I’m convinced that for large-scale web applications, Angular is not only the best solution, but the only scalable one that can be used no matter how large you foresee your application becoming.

React still has a place for smaller and mid-sized applications. In fact, in that category it outshines Angular because of its cross-platform portability through React Native and also its ease of use as a framework with a shorter learning curve. However, scalability challenges occur much sooner with React than with Angular. Most recently (and famously), Airbnb chose to sunset React Native because of (among other issues) the difficulty with refactoring React code, along with the relatively steep learning curve and boilerplate involved with Redux.

Honestly, if you think the learning curve with Redux is steep, the learning curve with Angular is even steeper. So in that respect, there isn’t a large advantage in using Angular. However, good programmers are known for focusing on long-term thinking as opposed to short-term. In the long run, once you get past the steep learning curve, here are a few ways that Angular 6 notably outshines and outperforms React.js, and why it’s worth learning despite how long it takes to pick up.

1. **Slick Tooling**

Like many languages and frameworks Google makes, Angular also has really slick tooling. Want to scaffold out a component? The Angular CLI makes it very easy to pinpoint exactly the folder you want to put it in, which module it should be imported into, and a basic unit test for making sure it renders properly. A karma test runner, a styling language of your choice, and now even adding custom templating languages like Pug is very easy to do with Angular 6. Being able to use Pug so easily

Angular 6 also adds two very important superpowers to the CLI: Schematics and Libraries. Schematics allow you to reduce your boilerplate code to a custom schematic that you can create for your own modules and components. Meanwhile, the new library generator makes it much easier to contribute open-source modules. Without it, I would still be largely working on closed-source projects rather than go through the hassle of making specific components very reusable and open-sourced. If it weren’t for the Angular 6 CLI and the convenience it provides, many developers like me who have their plates full already writing code for more than 60 hours per week, would probably not contribute to open source.

Finally, when you use Angular and Typescript with a good text editor like Visual Studio Code, your code gains magical abilities to automatically enter your thoughts and figure out which variable you are trying to use, automatically import it, and use it. You also get the ability to auto-complete your classes with their attached methods that you didn’t even know existed, import modules that you didn’t know were included, and just have to read documentation much less frequently because everything you need to read is right next to your code inside your editor.

**2. Type Safety that Leads to Easier Refactoring**

Typescript was my first statically typed OOP Language. There, I said it. Like many JavaScript developers, my background was not in formal Computer Science. My first Computer Science course was a Python class on Coursera. Naturally, Python isn’t statically typed either, and I liked it that way. Once I got comfortable with debugging, I felt very nimble in writing code that just worked in Python, especially for scripting and CSV reading.

Well, turns out that with JavaScript, something that “just works” in one situation often requires careful pre-planning and design beforehand to make sure it will also work in future situations as requirements change and products become more complex.

Almost every JavaScript developer makes this mistake of underestimating future changes to code at some point. Regardless, as I continued to dive deeper into JavaScript and became more proficient with React and Redux, I definitely appreciated the effort by the community to provide not only best practices for UI elements, but also state management. That’s probably the single best characteristic about React: the thriving developer community.

However, as I would refactor my react code, I’d often be stuck about when to return a reference to a method versus a function that returns that method versus calling that method in my class outright. For example, imagine we have a parent App Component calling a Child.

import React, {Component} from 'react';

import Child from './child';

export default class App extends Component {

handleClick(event) {

console.log('Child clicked from ' + event.target.value);

}

render() {

return (

<Child onClick={this.handleClick} />

)

}

}

In these cases, event will be undefined because we aren’t explicitly bubbling it up from the Child component. However, we can replace the line

<Child onClick={this.handleClick} /> with

<Child onClick={(event) => this.handleClick(event)}

and things would magically seem to work. Oh, except for binding ‘this’ to the handleClick method (eh, small detail). In the child, we would need to specifically reference the parent callback attached to props, so if the child was a simple button, it might look something like

export default const Child = (props) => {

return (

<button onClick={props.onClick}>Default text</button>

);

}

In order to handle situations like this, react used to come bundled with a handy library called… prop-types (now a separate install). In other words, react also acknowledged the importance of having types in your code to make things more sane and sort out subtle differences like these. However, unlike in React, types are a first-class construct in Angular with Typescript. Might as well bite the bullet and start adding types to our JavaScript code (especially for enterprise-scale apps).

At first it seemed tedious trying to learn Typescript and Angular at the same time, and definitely added to the learning curve of a new framework (yes, despite Angular being on version 6, it’s still a much newer framework than React since it’s nothing like Angular 1). However, after a few months of becoming familiar with the project you work with and easing into the not-quite-JS syntax going through a strict compiler, the ease of refactoring existing code is worth the trouble of adding types many times over because types only need to be added once, whereas refactoring is an ongoing process that occurs continuously over several years.

1. **Built-in Data Streaming**

Another pain point that new Angular developers face is that promises no longer seem to be first class citizens. What?!? No promises?!? Shocking. You can still use promises if you really want, but Angular installs RxJS for us, which is an asynchronous programming library that favors Observables over promises. At first it took some time

to wrap my head around Observables, and I’m not sure I’ve fully unlocked their capabilities still.

The main difference is that whereas promises allow us to listen to asynchronous data once, Observables allow us to continue listening for new data and make changes automatically as the data changes. Observables are also very easy to pass around and run several data-formatting operations on without even receiving the data.

With Angular 6, RxJS also received a big update and seems to be following Angular’s semantic versioning now. More importantly, RxJS 6 allows RxJS to be used in any JavaScript environment, with or without Angular. After transitioning to Observables and away from promises, I’m convinced they give any software developer much more flexibility around dealing with Asynchronous code.

Streams are a very natural way to think about data flow, whereas promises felt a bit out of place and felt like they were not powerful enough to handle more real-time functionality like notifications, instant changes based on user actions, among other things. Compound that with the large scale of enterprise applications where, often, two large modules need some bridge to communicate specific pieces of information with each other. RxJS, when used properly and thoughtfully, allows you to build strong, durable bridges across different modules of your Angular application.

**Conclusion**

Built-in data streaming, type safety, and a modular CLI are hallmarks of what makes Angular such an opinionated framework. However, considering that this is the framework that helps even the largest of technology companies manage their enterprise applications, combined with a commitment from Google to introduce no more breaking changes, now is a better time than ever to learn Angular.

Be warned, there is at least a few months of learning involved before you feel proficient enough to start building non-static content, but it’s worth it. When I think about how painless it was to upgrade Angular versions versus how painful it is to upgrade React dependencies, that alone makes the switch more productive for me.

Never get stuck with one language in IT, and never with any JS. The world of JS is very unstable and changing, evolving rapidly. A few years ago we had Jquery, then we started with ember, then angular, then react giving a tough competition. Angular 1.x is no longer the seasoned skill in the market.

Mostly the companies working on angular 1.x have moved to either Angular 2.x with Typescript or ReactJS. There have come several other JS variants in the market by

now and I am not in touch with JS for last few months and therefore, I feel like lagging an era now!

There are so much improvements, new packages, new libraries on JS coming every day. In order to be a ready-to-make developer, you need to be aware what’s changing around you. There has been so much manipulating the DOM and not-manipulating the DOM experiments with JS, from interpreters to transpilers, from running on client side to server side, from backward compatibility to forward compatibility, from OOP to Functional paradigm and switching back to the previous one—— All these experiments result

* **Importance and Applicability**

Angular is an open-source front-end framework developed by Google for creating dynamic, modern web apps. First introduced in 2009, the framework has gained huge traction over the years for eliminating unnecessary code and ensuring lighter & faster apps.

Having rapidly evolved from AngularJS in 2010 to Angular 5 in 2017, the front-end framework is today used by more than 44.3% of software engineers for creating user interfaces (Stack Overflow Developer Survey 2017).

Angular helps build interactive and dynamic single page applications (SPAs) with its compelling features including templating, two-way binding, modularization, RESTful API handling, dependency injection, and AJAX handling. Designers can use HTML as template language and even extend HTML’ syntax to easily convey the components of the application.

You don’t need to rely on third-party libraries to build dynamic applications with Angular. Using this framework in your projects, you can reap multiple benefits. We have listed the reasons below:

* **Supported by Google**

One of the biggest advantage of Angular is that it is supported by Google. The best part about it is Google’s Long-Term Support (LTS). This sheds light on Google’s plan to stick with it and further scale up the Angular ecosystem.

Google apps also use this framework and their team is very optimistic about Angular’s stability. Other Angular developers also get a golden opportunity to learn from certified professionals.

* **TypeScript**

Angular applications are built using TypeScript language, a superscript for JavaScript, which ensures higher security as it supports types (primitives, interfaces, etc.). It helps catch and eliminate errors early when writing the code or performing maintenance tasks.

Unlike CoffeeScript or Dart, TypeScript is not a stand-alone language. With TypeScript, you can easily take the existing ES5 or ES2015+ JS code and it will compile it down based on what you are configuring. It fully supports core ES2015 and ES2016/ES2017 features such as decorators or async/await.

You can directly debug TypeScript code in the browser or an editor if you have proper map files created during build time. This language ensures improved navigation, refactoring, and autocompletion services. You can even opt out of its inbuilt features when needed.

* **Declarative UI**

Angular uses HTML to define the UI of the application. HTML, as compared to JavaScript, is a less convoluted language. HTML is also a declarative and intuitive language.

How does it help? You don’t need to invest your time in program flows and in deciding what loads first. Define what you require and Angular will take care of it.

* **POJO**

With Angular, you don’t need any additional getter and setter functions. Since, every object it uses is POJO (Plain Old JavaScript Object), which enables object manipulation by providing all the conventional JavaScript functionalities. You can remove or add properties from the objects, while also looping over these objects when required.

* **Easy Testing**

In Angular, testing is extremely simple. Angular.js modules has the application parts, which are easy to manipulate. With module separation, you can load the necessary services, while effectively performing automatic testing. You don’t even need to remember module loading order if you follow “one file-one module” principle.

* **Simplified MVC Pattern**

Angular framework is embedded with original MVC (Model-View-Controller) software architectural setup. However, it is not according to the established standards. Angular does not ask developers to split an application into different MVC components and build a code that could unite them.

Rather, it only asks to divide the app and takes care of everything else. Therefore, Angular and MVVM (Model-View-View-Model) design structure are quite similar.

Angular ensures easy development as it eliminates the need for unnecessary code. It has a simplified MVC architecture, which makes writing getters and setters needless. Directives can be managed by some other team, as these are not part of app code. All in all, developers are promised less coding, along with lighter and faster apps. And according to Amazon, every 100-millisecond improvement in page loading speed led to 1% increase in revenue.

* **Modular Structure**

Angular organizes code into buckets, whether it is components, directives, pipes, or services. Those who are familiar with Angular refer to these buckets as modules. Modules make application functionality organization easy, segregating it into features and reusable chunks. Modules also allow for lazy loading, which paves way for application feature loading in the background or on-demand.

Angular makes it an achievable goal to divide the labor across different team members while ensuring organized code. You can make the best of modules when you have a proper understanding of these. Developers can improve productivity with appropriate modules built.

* **Code Consistency**

Any code base requires consistent coding. A writer knows how important consistency is in their content pieces. We know if the content fails to resonate with the readers to a deeper level at any touchpoint, we are on a downward slope of lead conversions. Coding is no different.

Inconsistent coding increases the risks of delayed launches or elevated costs. Unlike it, consistent coding has several benefits, such as it makes sites easier to use and enables the use of templates or pre-defined code snippets.

Angular framework is based on components, which begin in the same style. For instance, each component places the code in a component class or defines a @Component decorator (metadata). These components are small interface elements independent of each other, and thus, offer you several benefits, including:

* + **Reusability**

The component-based structure of Angular makes the components highly reusable across the app. You can build the UI (User Interface) with moving parts, while also ensuring a smooth development process for developers.

* + **Simplified Unit-Testing**

Being independent of each other, the components make unit testing much easier.

* + **Improved Readability**

Consistency in coding makes reading the code a piece of cake for new developers on an ongoing project, which adds to their productivity.

* + **Ease of Maintenance**

Decoupled components are replaceable with better implementations. Simply put, it enables efficient code maintenance and update.

* **Role and profile**

**Frontend developer in Angular: -** Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that data.

* **Tools used for front-end development**

There are several tools and platforms (wordpress, magento etc..) available that can be used to develop the front end of a website, and understanding which tools are best fit for specific tasks marks the difference between developing a hacked site and a well designed, scalable site

* **Hyper Text Markup Language (HTML)**

Hyper Text Markup Language (HTML) is the backbone of any website development process, without which a web page doesn't exist. Hypertext means that text has links, termed hyperlinks, embedded in it. When a user clicks on a word or a phrase that has a hyperlink, it will bring another web-page. A markup language indicates text can be turned into images, tables, links, and other representations. It is the HTML code that provides an overall framework of how the site will look. HTML was developed by Tim Berners-Lee. The latest version of HTML is called HTML5 and was published on October 28, 2014 by the W3 recommendation. This version contains new and efficient ways of handling elements such as video and audio files.

* **Cascading Style Sheets (CSS)**

Cascading Style Sheets (CSS) controls the presentation aspect of the site and allows your site to have its own unique look. It does this by maintaining style sheets which sit on top of other style rules and are triggered based on other inputs, such as device screen size and resolution.

* **JavaScript**

JavaScript is an event-based imperative programming language (as opposed to HTML's declarative language model) that is used to transform a static HTML page into a dynamic interface. JavaScript code can use the [Document Object Model] (DOM), provided by the HTML standard, to manipulate a web page in response to events, like user input.

Using a technique called AJAX, JavaScript code can also actively retrieve content from the web (independent of the original HTML page retrieval), and also react to server-side events as well, adding a truly dynamic nature to the web page experience.

* **WebAssembly**

WebAssembly, supported by all the major browsers (i.e. from the major vendors Google, Apple, Mozilla and Microsoft), is the only alternative to JavaScript for running code in web browsers (without the help of plug-ins, such as Flash, Java or Silverlight; all being discontinued, as browsers are dropping plug-in support). Prior to its adoption, there was asm.js (a subset of JavaScript; and thus strictly works in all browsers), that's also used as a compiler target with efficient support in browsers such as Internet Explorer 11; and for such browsers that do not support WebAssembly directly, it can be compiled to asm.js and those browsers supported that way. Generally speaking programmers do not program in WebAssembly (or asm.js) directly, but use languages such as Rust, C or C++ or in theory any language, that compile to it.



**CHAPTER- 3**

# BRIEF DESCRIPTION OF THE WORK DONE

* **Position of Internship and roles**
* **Frontend developer in Angular: -** Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, Angular so that users can view and interact with that data.
* Front End Developers are computer programmers who specialize in website design. Front End Developer duties include determining the structure and design of web pages, striking a balance between functional and aesthetic design and ensuring web design is optimized for smartphones.
* We are looking for programmers with a keen eye for design for the position of Front End Developer. Front end Developers are responsible for ensuring the alignment of web design and user experience requirements, optimizing web pages for maximum efficiency and maintaining brand consistency across all web pages, among other duties.
* Front End Developers are required to work in teams alongside Back end Developers, Graphic Designers and User Experience Designers to ensure all elements of web creation are consistent. This requires excellent communication and interpersonal skills.

**Front End Developer Responsibilities:**

1. Determining the structure and design of web pages.
2. Ensuring user experience determines design choices.
3. Developing features to enhance the user experience.
4. Striking a balance between functional and aesthetic design.
5. Ensuring web design is optimized for smartphones.
6. Building reusable code for future use.
7. Optimizing web pages for maximum speed and scalability.
8. Utilizing a variety of markup languages to write web pages.
9. Maintaining brand consistency throughout design.

**Front End Developer Requirements:**

1. Degree in Computer Science or related field.
2. Understanding of key design principles.
3. Proficiency with HTML, CSS, JavaScript and jQuery.
4. Understanding of server-side CSS.
5. Experience with graphic design applications such as Adobe Illustrator.
6. Experience with responsive and adaptive design.
7. Understanding of SEO principles.
8. Good problem solving skills.
9. Excellent verbal communication skills.
10. Good interpersonal skills.

* **Activities/ equipment handled**

I was working using the following tools to do the task:

* **VS CODE: -**

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).

In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents claiming to use it.

Visual Studio Code is a source code editor that can be used with a variety of programming languages. Instead of a project system it allows users to open one or more directories, which can then be saved in workspaces for future reuse.

This allows it to operate as a language-agnostic code editor for any language, contrary to Microsoft Visual Studio which uses the proprietary .sln solution file and project-specific project files. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many of Visual Studio Code features are not exposed through menus or the user interface, but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, perform static code analysis, add code linters, using the Language Server Protocol and connect to additional services.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character for Windows/Linux, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

Language support

Out-of-the-box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

* **NODE JS: -**

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

Though .js is the standard filename extension for JavaScript code, the name "Node.js" does not refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games)

npm is the pre-installed package manager for the Node.js server platform. It installs Node.js programs from the npm registry, organizing the installation and management of third-party Node.js programs. Packages in the npm registry can range from simple helper libraries such as Lodash to task runners such as Grunt.

* **Angular CLI: -**

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications. You can use the tool directly in a command shell, or indirectly through an interactive UI such as Angular Console.

The ng new command creates an Angular workspace folder and generates a new app skeleton. A workspace can contain multiple apps and libraries. The initial app created by the ng new command is at the top level of the workspace. When you generate an additional app or library in a workspace, it goes into a projects/ subfolder.

A newly generated app contains the source files for a root module, with a root component and template. Each app has a src folder that contains the logic, data, and assets.

You can edit the generated files directly, or add to and modify them using CLI commands. Use the ng generate command to add new files for additional components and services, and code for new pipes, directives, and so on. Commands such as add and generate, which create or operate on apps and libraries, must be executed from within a workspace or project folder.

A single workspace configuration file, angular.json, is created at the top level of the workspace. This is where you can set per-project defaults for CLI command options, and specify configurations to use when the CLI builds a project for different targets.

* **Challenges faced and how those were tackled**

Biggest challenge faced was not having clarity about the components involved as it involves a lot of business-related terms to be understood. It took me time to understand them and take decision. I was helped greatly by my manager and people on site to get through them.

Challenge related to forecasting chain was to go through these many scripts and java programs without having much idea about anything of the system. So I started small looked at the scripts which seemed to have lesser impact, the scripts which had basic task of sending the feed to the external service provider.

From there on I was a bit comfortable in the territory I started to look into other scripts which were doing more complicated tasks. The best thing which came to rescue were notes which I made about each and every step. The challenge with the notes was also that they had to be descriptive and also short as they had to be looked into by managers on on-site.

The biggest hurdle in the analysis became a job stream which was very tightly coupled and the plan was to break it down. It involved about scripts which were using its predecessor’s data and also its successor’s data on its next run. There were a lot of discussions and debates on how the data flows and is being used and how further this should be dealt with.

There were some things like I had to learn, like how to store data without any database how to make a website using just only angular, I had to figure its interface out well enough so that I can use it well.

With the task helping in retire the database, which was the second task I had in Parts Supply Chain, I had to do a lot of thinking and cognitive analysis to move forward. It was something which we were looking from a new end as technical people who had worked on it had left. The document which we had wasn’t too descriptive to help us out and the system in itself was very complicated. We didn’t have a clear beginning and an ending.

We had to make assumptions which sometimes turned out to be wrong and a lot of time we were on correct path with the assumptions. The assumption which had made final, we had a lot of fear that it may fall apart and we might have to look it with a completely new point of view till the very end of the analysis. But, at the end it held upright and we were able to close it off.

Challenges are tackled by me by facing every challenge with positive attitude and determination and by focusing. I always yes to learning new things without any hesitation and fear. To achieve success in every project in which I had worked my colleagues and my mangers always support me and they help me get out of the problems.

* **Learning outcomes**

In this internship I have learned Angular 8, Bootstrap 4, HTML, CSS and for backend I have installed package.json for json file which acts as backend server and db. Json file acts as database storage.

* **Angular: -** AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. (This flexibility has led to the acronym MVW, which stands for "model-view-whatever" and may also encompass model–view–presenter and model–view–adapter.) In 2014, the original AngularJS team began working on the Angular web framework.
* **Bootstrap: -** Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

Bootstrap 4 is the newest version of Bootstrap, which is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

Bootstrap 4 is completely free to download and use

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

Example of a webpage using Bootstrap framework

Example of a webpage using Bootstrap framework rendered in Firefox

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the four predefined fixed widths, depending on the size of the screen showing the page:

A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap, however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled Sass files.

* **HTML: -** Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents 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. HTML 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 /> directly introduce content into the page. Other tags such as <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 affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

* **CSS: -** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

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

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.



**CHAPTER- 4**

**CARS MANAGEMENT PROJECT**

* **In****troduction**

Cars project is the project that is made by me in the internship in my organisation.

This project is made by angular only. In this project there is json file which act as backend server for the project so this project is the full project in which frontend and backend both are available but the backend is without any server and any backend languages.

Car management system where we can find the details of the car and we also can edit, add, delete car dynamically. We can find the car which has highest price amongst all

* **Languages**

In this project I have used Angular 8, Bootstrap 4, HTML, CSS and for backend I have installed package.json for json file which acts as backend server and db. Json file acts as database storage.

* **Angular: -** AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. (This flexibility has led to the acronym MVW, which stands for "model-view-whatever" and may also encompass model–view–presenter and model–view–adapter.) In 2014, the original AngularJS team began working on the Angular web framework.
* **Bootstrap: -** Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

Bootstrap 4 is the newest version of Bootstrap, which is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

Bootstrap 4 is completely free to download and use

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

Example of a webpage using Bootstrap framework

Example of a webpage using Bootstrap framework rendered in Firefox

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the four predefined fixed widths, depending on the size of the screen showing the page:

A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap, however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled Sass files.

* **HTML: -** Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents 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. HTML 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 /> directly introduce content into the page. Other tags such as <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 affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

* **CSS: -** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

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

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

* **Working Structure of the project**

The Angular CLI ng new command creates a workspace.

ng new <PROJECT-NAME>

When you run this command, the CLI installs the necessary Angular npm packages and other dependencies in a new workspace, with a root-level application named my project. The workspace root folder contains various support and configuration files, and a README file with generated descriptive text that you can customize.

By default, ng new creates an initial skeleton application at the root level of the workspace, along with its end-to-end tests. The skeleton is for a simple Welcome application that is ready to run and easy to modify. The root-level application has the same name as the workspace, and the source files reside in the src/ subfolder of the workspace.

* **Files in the project**
  + **Components: -** There are four components Cars, Dashboard, Report and cars-add. Cars component in which we can add edit delete any car and it will instantly render into the database and all the data get updated instantly. Report component is where we can find the details of the cars. Cars-add component help us to add cars dynamically in our project. Dashboard component is the component which is shown to the user very first.
  + **Services: -** The services will help us to apply same functionality to multiple component without writing the same functionality again and again. In this project there is one service that is car-service in which functions are there by which we can dynamically update our database files, add car, delete car, edit car.
  + **Modal: -** Modals are the folder in which ng class is there which provide structure to our form and as well as to our json file in that format our data is storing in json file.
  + **Routing: -** app.routing.module file provide the url path to different components so that our project’s component should navigate to proper and correct url.
  + **Module:-** app.module.ts file provide essentials packages and modules that is necessary to our project.
* **Output of the project**

The project output is working properly without any compile time waring and error.



**CHAPTER- 5**

# CONCLUSION

So far it has been a great learning experience at eClerx Services Limited. Currently, I am excited over working on new tasks assinged which involve upgradation of web services to new technologies. Getting involved in critical tasks has taught me a lot and provided knowledge on how to approach problems and if stuck how to come up with a solution.

As there is a shift towards cloud technologies, in future I see getting myself exposed to more technologies and nourish my technical skills more. I also want to be a full time employee at eClerx Services Limited after completion of my internship. I am very excited for the coming future.

**REFERENCES**

1. https://www.w3schools.com/bootstrap/
2. https://www.w3schools.com/css/default.asp
3. https://www.w3schools.com/html/default.asp
4. https://angular.io/
5. https://material.angular.io/
6. https://getbootstrap.com/docs/4.4/getting-started/introduction/
7. https://jsonplaceholder.typicode.com/

1. https://en.wikipedia.org/wiki/Node.js
2. https://nodejs.org/en/

## Annexure-XI

**Training Letter Format**

**RefNo: LPU/ Dated: / /**

**To,**

**---------------------------------- (CEO/GM/HR Manager),**

**---------------------------------- (Company Name),**

**--------------- (State)**

**Subject: Internship for**(Program Name) **student from LPU**

Dear Sir/Madam,

Lovely Professional University has been a front-runner to improve standards of higher education in the country. The University, through its innovative pedagogical interventions, focuses on enhancing experiential learning with support and engagement of the industry.

|  |
| --- |
| (Name of the School) |

Thegrooms the student through analytical teaching, live projects, and active interface with industry professionals. As part of curricular requirements, the students are expected to undergo an internship in industry for gaining exposure to corporate practices and work environment.

We shall be grateful if you provide an internship opportunity in your esteemed organization to

(Student Name)bearing Registration Number for (duration), where the internship is expected to end before DD/MM/YY.

We are confident that the student will contribute to value creation and meet your expectations should he/she get an opportunity for internship.

We look forward to a favorable response.

Best Regards,

(Name)**Contact Number:**

**Training Coordinator,**

**Official Email:**

(Name of the School),

**Lovely Professional University, Punjab**