**WISH LAB**

by

Aditya Singh (1516410017)

Akanksha Mishra (151641022)

Ayushi Dwivedi (1516410079)

Ekta Tiwari (1516410097)

Ritik Srivastava (1516410)

Rohit Raj Rai (1516410223)

Submitted to the Department of Computer Science and Engineering

in partial fulfillment of the requirements

for the degree of

**BACHELOR OF TECHNOLOGY**

**(COMPUTER SCIENCE & ENGINEERING)**



Dr. APJ Abdul Kalam Technical University, Lucknow

May, 2019

**TABLE OF CONTENTS Page**

DECLARATION ...…………………………………………………………………… ii

CERTIFICATE ...…………………………………………………………………....... iii

ACKNOWLEDGEMENTS ….……………………………………………………… iv

ABSTRACT ..………………………………………………………………………... v

LIST OF TABLES ……………………………………………………………………. vii

LIST OF FIGURES…………………………………………………………………... viii

LIST OF SYMBOLS…………………………………………………………………. ix

LIST OF ABBREVIATIONS………………………………………………………… x

CHAPTER 1 INTRODUCTION……………………………………………………... 1

1.1 EARLY SYSTEMS………………………………………………………….. 2

1.2 GROWTH DURING 19th CENTURY………………………………………. 3

1.3 IN MODERN ERA………………………………………………………...... 4

1.4 CHALLENGES FACED BY NON-PROFIT OR THIRD SECTOR

ORGANIZATIONS…………………………………………………………... 5

CHAPTER 2 PROBLEM STATEMENT……………………………………………. 6

CHAPTER 3 DESIGN ………………………………………………………………. 7

CHAPTER 4 PLANNING …………………………………………………………... 8

CHAPTER 5 CODING ……………………………………………………………… 9

CHAPTER 6 TESTING ……………………………………………………………... 10

CHAPTER 7

***DECLARATION***

*I hereby declare that this submission is my own work and that to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.*

*Signature : Signature:*

*Name : Name :*

*Roll No. : Roll No. :*

*Date : Date :*

*Signature Signature*

*Name Name*

*Roll No. Roll No.*

*Date Date*

*Signature Signature*

*Name Name*

*Roll No. Roll No.*

*Date Date*

**CERTIFICATE**

This is to certify that Project Report entitled “Wish Lab” which is submitted by Aditya Singh, Akanksha Mishra, Ayushi Dwivedi, Ekta Tiwari, Ritik Srivastava, Rohit Raj Rai in partial fulfillment of the requirement for the award of degree B. Tech in Department of Computer Science & Engineering of U. P. Technical University, is a record of the candidate own work carried out by him under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Signature of Guide

**ACKNOWLEDGEMENT**

*It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech Final Year. We owe special debt of gratitude to Professor Anshuman Singh, Department of Computer Science & Engineering, College of Engineering, Lucknow for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.*

*We also take the opportunity to acknowledge the contribution of Professor M. S. Dhoni, Head, Department of Computer Science & Engineering, College of Engineering, Lucknow for his full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.*

*Signature:*

*Name :*

*Roll No.:*

*Date :*

*Signature:*

*Name :*

*Roll No.:*

*Date :*

***ABSTRACT***

*Wish Lab is a web based application that intends to socially fulfill the basic requirements of vagrants, poor, orphans, needy etc. It aims to create an environment where people willingly extend their support to help someone in need and no place other than a multi-national organization can help achieve this aim in a better way. This not only helps**the needy but also helps to experience a sense of satisfaction and happiness when their wishes are fulfilled.*

*Since it is quite impractical in this busy world to look after the requirements of vagrants so our Wish Lab application helps to keep track and find the best possible way to fulfill someone’s wishes.*

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNo.** | **Table Name** | **Description** | **Primary Key** |
| **1.** | Wishlist | This table contains all the required wishes. | Wishes |
| **2.** | Picked Wish | This table contains all the wishes picked by the employee. | WishId |
| **3.** | Fulfilled Wish | It contains all the wishes fulfilled by several employees. | FId |
| **4.** | Employee | It contains all the basic details of employees. | EmpId |
| **5.** | Admin | It contains all the basic details of admin. | AdminId |
| **6.** | Volunteer | It contains all the basic details of volunteers. | VId |

**LIST OF FIGURES**

**1. Service Layer :** Service layer is the layer in which the business logic of enterprise application is implemented. It is the heart of an enterprise application as it mirrors the business processes of the organization. It interacts with the presentation layer and persistence layer. In this course we will deep dive into development of **Service Layer** of an enterprise application using **Spring Framework**.

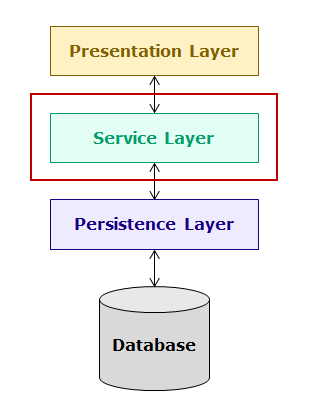


Figure 1. Service Layer

**1.1. Developing Service Layer :**

For implementing Admin Login user story we will create following classes:

* **AdminLoginServiceImpl** – The business logic of this user story is implemented in this class. It interacts with the AdminLoginController in presentation layer and AdminLoginDAOImpl class in persistence layer.
* **AdminLoginDAOImpl** – This class interacts with the database and AdminLoginServiceImpl class of service layer.
* **AdminLoginController** – This class will interact with the user interface and sends the data to AdminLoginServiceImplclass of service layer.
* **AdminLogin**– This is model class which is used to transfer data across different layers.

When customer enters the login credentials the data will flow from AdminLoginController to AdminLoginServiceImpl and then to AdminLoginDAOImpl class as follows:

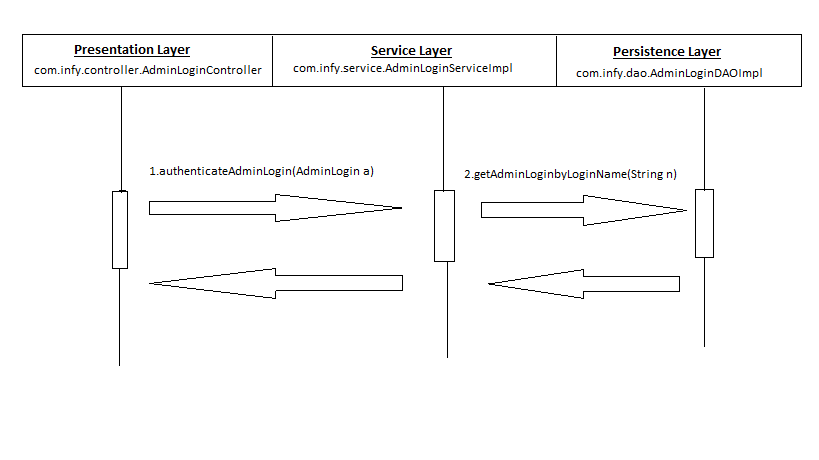


Figure 1.1.1 Developing Service Layer for Admin Login

For implementing Employee Login user story we will create following classes:

* **EmployeeLoginServiceImpl** – The business logic of this user story is implemented in this class. It interacts with the EmployeeLoginController in presentation layer and AdminLoginDAOImpl class in persistence layer.
* **EmployeeLoginDAOImpl** – This class interacts with the database and EmployeeLoginServiceImpl class of service layer.
* **EmployeeLoginController** – This class will interact with the user interface and sends the data to EmployeeLoginServiceImplclass of service layer.
* **EmployeeLogin**– This is model class which is used to transfer data across different layers.

When customer enters the login credentials the data will flow from EmployeeLoginController to EmployeeLoginServiceImpl and then to EmployeeLoginDAOImpl class as follows:

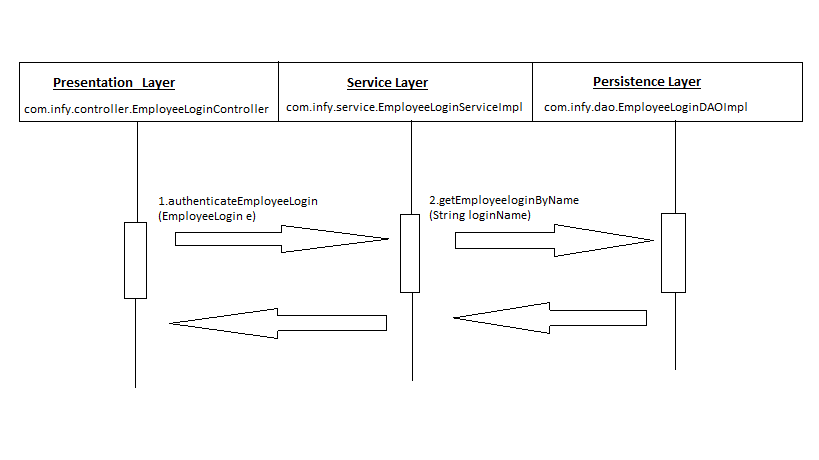


Figure 1.1.2 Developing Service Layer for Employee Login

For implementing Volunteer Login user story we will create following classes:

* **VolunteerLoginServiceImpl** – The business logic of this user story is implemented in this class. It interacts with the VolunteerLoginController in presentation layer and VolunteeerLoginDAOImpl class in persistence layer.
* **VolunteerLoginDAOImpl** – This class interacts with the database and VolunteerLoginServiceImpl class of service layer.
* **VolunteerLoginController** – This class will interact with the user interface and sends the data to VolunteerLoginServiceImplclass of service layer.
* **VolunteerLogin**– This is model class which is used to transfer data across different layers.

When customer enters the login credentials the data will flow from VolunteerLoginController to VolunteerLoginServiceImpl and then to VolunteerLoginDAOImpl class as follows:

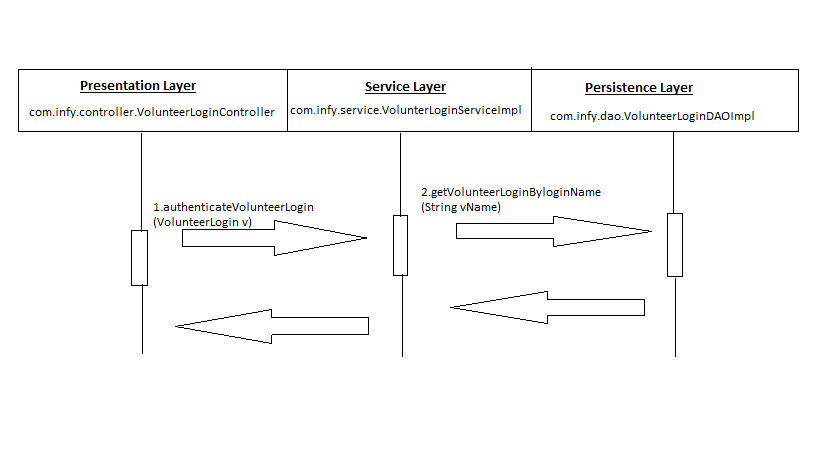


Figure 1.1.3 Developing Service Layer for Volunteer Login

**1.2. Spring IoC Container :**

Spring framework provides a container for dependency injection. This container is used to create, initialize, and inject required objects. The classes whose objects life cycle is managed by Spring are called as **beans** or **Spring beans**. Spring container is also called as **Spring** **IoC container**. The org.springframework.beans and org.springframework.context packages are the basis for Spring IoC container.

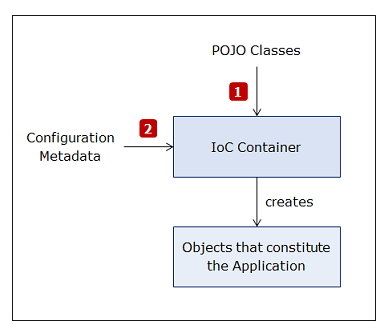
****

Figure 1.2 Spring IoC Container

The Spring IoC container is represented by following interfaces:

* The **BeanFactory**interface represents the basic container which provides basic functionalities. It instantiates bean whenever asked for by the client application. Using its **getBean() method**  you can get instances of beans.
* The **ApplicationContext**interface extends BeanFactory interface and provides additional functionalities to support enterprise application development. There are many implementations of this interface. Some commonly used implementation classes are as follows:
  + **ClassPathXmlApplicationContext**: It is used for Java applications using XML based configuration.
  + **AnnotationConfigApplicationContext**: It is used for Java applications using annotations based configuration.

**2. Persistence Layer :** **Persistence Layer** of an enterprise application contains the logic to interact with the database using **Hibernate Framework.**

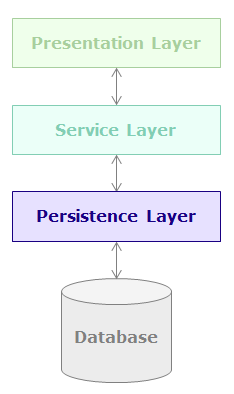
****

Figure 2. Persistence Layer

**3. Data Persistence Technologies :** There are many technologies which are used to develop the persistence layer. The following diagram shows some of those technologies:

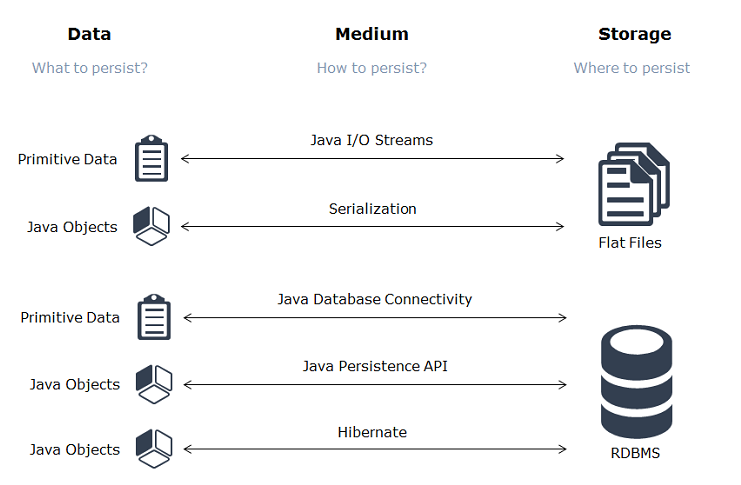
****

Figure 3. Data Persistence Technologies

**4. Object Relational Mapping (ORM) Frameworks :** Object Relational Mapping (ORM) frameworks provide a way for accessing a relational database from an object-oriented language. With the help of ORM frameworks, developers can map Java classes to tables using metadata.

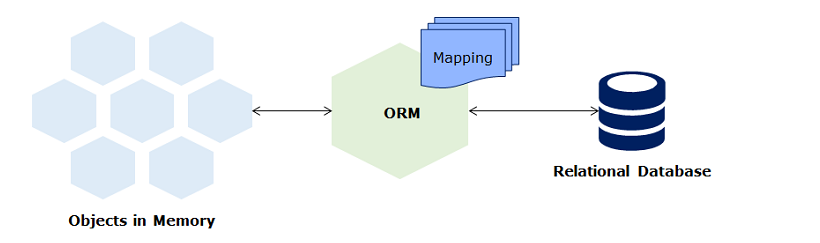
****

Figure 4. ORM Frameworks

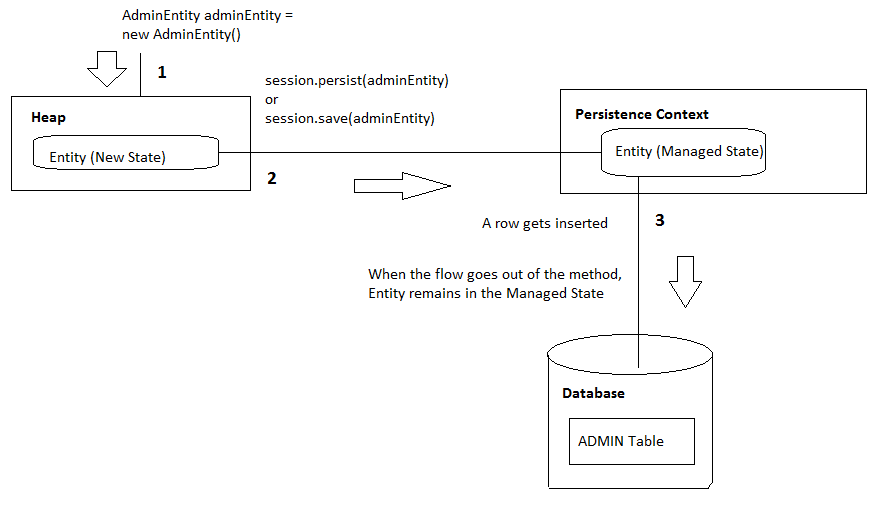
**5. Entity Lifecycle :** The different states of a persistent object are as follows:

* **Transient State**: A newly created object which is not associated with a persistence context.
* **Persistent State**: An entity instance with a row representation in the database and it is currently associated with a persistence context.
* **Detached State**: An entity instance with or without a row identity in the database and which is no longer associated with a persistence context, usually because the persistence context was closed or the instance was evicted from the context.
* **Removed State**: An entity instance with a row identity in the database and associated with a persistence context, but scheduled for removal from the database.

**5.1 State Transition During Insert Operation :**

During insert operation entity object will go through following states:

1. Entity will be in **new state** after its creation.
2. State transition for the entity will happen from **new to managed** state when persist() or save() is called.
3. When the flow goes out of the method, a new row will be inserted in the table. Also, the entity continues to be in **managed state**.

Figure 5.1 Insert Operation

**5.2 State Transition During Update Operation :**

During update operation entity will go through following states:

1. Entity will be retrieved in **managed state** when get() method is invoked.
2. On updation the entity remains in the managed state.
3. When the flow goes out of the method, changes will get reflected in the corresponding row of the table. Also, the entity continues to be in **managed state**.

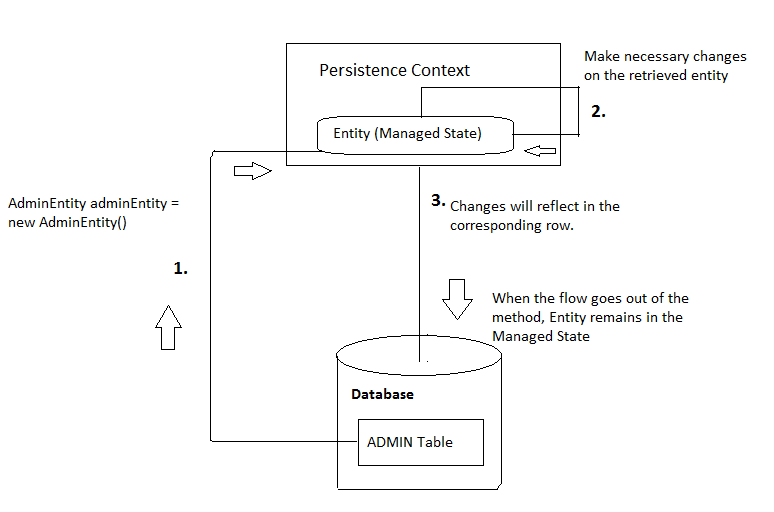


Figure 5.2 Update Operation

**5.3 State Transition During Delete Operation :**

During delete operation entity object will go through following states:

1. Entity will be retrieved in **managed state** when get() method is invoked.
2. State transition for the entity will happen from managed to **removed state** when delete() is done.
3. When the flow goes out of the method, corresponding row will get deleted from the table. Also, the entity moves to a new state.

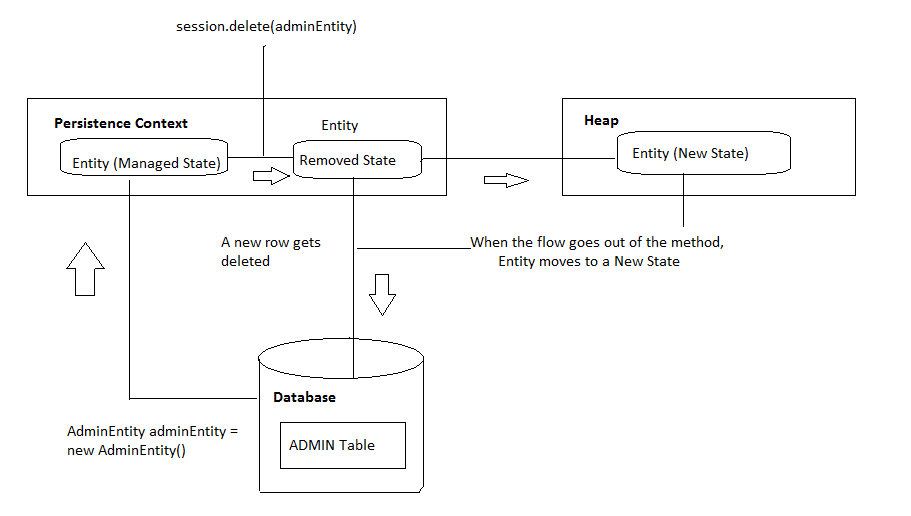


Figure 5.3 Delete Operation

**LIST OF SYMBOLS**

**LIST OF ABBREVIATIONS**

**CHAPTER 1**

**INTRODUCTION**

1. **Introduction:**

It is an humanitarian act to take care of someone and try to fulfill his or her needs. This act is popularly called as charity. **Charity** is the act of giving help to those in need of it. It involves giving money, goods or time and effort to those who need it. It is done without expecting something in return. Giving money or food to poor people is an example of charity. Poor, sick or injured people are generally considered the proper people to whom

charity should be given. When such people are not supported, they often begin begging, which is directly asking for help from people they do not know.

* 1. **Early Systems :**

Until the mid-18th century, charity was mainly distributed through religious structures (such as the English Poor Laws of 1601), almshouses and bequests from the rich. Both Christianity and Islam incorporated significant charitable elements from their very beginnings[[4]](https://en.wikipedia.org/wiki/Charitable_organization#cite_note-4) and *dāna* (alms-giving) has a long tradition in Hinduism, Jainism, Buddhism and Sikhism. Charities provided education, health, housing and even prisons. Almshouses were established throughout Europe in the Early Middle Ages to provide a place of residence for poor, old and distressed people; King Athelstan of England (reigned 924-939) founded the first recorded almshouse in York in the 10th century.

**1.2 Growth During 19th Century :**

During the 19th century a profusion of charitable organizations emerged to alleviate the awful conditions of the working class in the slums. The Labourer's Friend Society, chaired by Lord Shaftesbury in the United Kingdom in 1830, aimed to improve working-class conditions. It promoted, for example, the allotment of land to labourers for "cottage husbandry" that later became the allotment movement. In 1844 it became the first Model Dwellings Company - one of a group of organizations that sought to improve the housing conditions of the working classes by building new homes for them, at the same time receiving a competitive rate of return on any investment. This was one of the first housing associations, a philanthropic endeavour that flourished in the second half of the nineteenth century brought about by the growth of the middle class. Later associations included the Peabody Trust(originating in 1862) and the Guinness Trust (founded in 1890). The principle of philanthropic intention with capitalist return was given the label "five per cent philanthropy.

2

**1.3 In Modern Era :**

In today’s busy scenario it’s a challenging task to know the needs of any needy person and fulfill them. It seems that we are now genuinely living through an era where disruptive change is happening faster than ever, and in which public confidence in charities is at a fairly low ebb.

The public is more sceptical and there’s a thirst for more scrutiny and transparency and a desire to understand how charities work and if things have gone wrong, why have they gone wrong. That is because there’s been a series of high profile, very problematic, very negative stories that have been played out in public over the last few years. Collectively as a sector, we need to respond to that.

We can’t just not do anything. There are people who need us today and people who will need us tomorrow. Last year Turn2Us helped five million people get information and access when they were struggling. We’ve got to keep going. We have to collectively ask, are there improvements, are there learnings from this?

**1.4** **Challenges Faced by Non-Profit or Third Sector Organization :**

We have a shiny new government. It is ready to give the voluntary sector the big bearhug of an embrace and the big wet kisses of an over-enthusiastic relative. The government also has its own agenda of the ‘big society’. But before we get carried away with the government’s agenda of what it wants to sort out – what are the issues that the sector needs to sort out. What are the big tasks that the sector needs to tackle to move onto the next stage of its development, to become an even stronger force in society today – indeed to play a full role in the big society? The list below is neither definitive nor exhaustive but based on nfpSynergy’s research with both charities and the public.

**Challenge 1: Growing income:**

Raising funds and increasing income is one of the biggest challenges for the sector as a whole. There is hardly an organisation which does not want to grow its income; which could not do more if it had more money. Our ‘State of the Sector’ survey completed by CEOs and managers from a breadth of organisations in 2009 put this as the top of the list of challenges that organisations faced. Put at its most simple, there are three ways that organisations can raise more money. They can be given it (voluntary donations), they can earn it (through delivering public services or running shops or social enterprises) or they can be given a grant (of the kind that BIG specialises in).

3

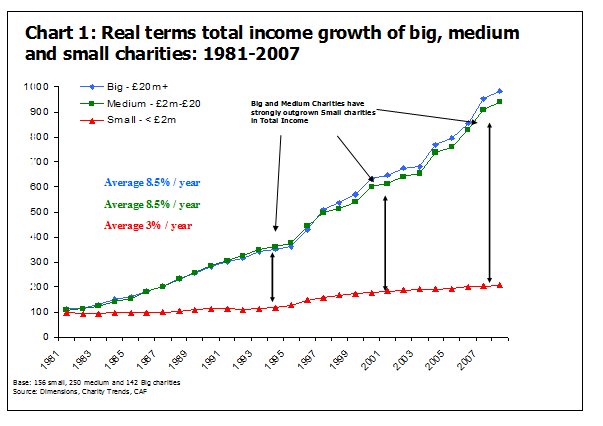
The real challenge in raising funds is to develop income streams which do not suffer from the ‘fallacy of composition’, as development economists’ call it. This is a solution that works for one organisation but is no good if everybody does the same thing.

So how will charities continue growing in size over the next 10-20 years is my first challenge for the sector?

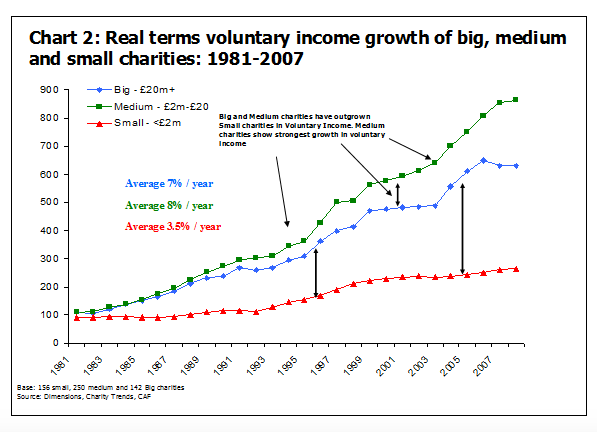
## Challenge 2: Small charities struggle to keep up

Charts 1 & 2 below show the average total income and voluntary income growth rates of small (up to £2m), medium (£2-£10 million) and large (over £10 million) charities over the last 30 years. Contrary to popular convention it is not medium-sized organisations that struggle but small ones. The question that this data raises comes in two parts. Firstly why does it happen? And secondly what can be done about it?

The difficulty that small charities have in growing is probably a bundle of factors. They can less afford to take risks because they don’t have the reserves if things go wrong. They don’t have the brand names that persuade people to give. The trustees probably are less comfortable with the type of professional fundraising that will raise income. Most sources of funding for small organisations are for restricted income, and so on.



4



As for the solution, these are even more guesswork (as we have never researched this kind of issue at nfpSynergy) but one survey we did a couple of years ago offers some clues. We asked smaller organisations how much unrestricted income they would trade if somebody had offered them a grant of £1 million in restricted income. In other words how much more valuable was unrestricted income compared to restricted income? A substantial minority of organisations would have accepted £400k of unrestricted income instead of £1 million in restricted income. The higher the percentage of restricted income that an organisation has, the less freedom it has to develop its unrestricted sources or to innovate and grow its brand.

## 5

## Challenge 3: Demonstrating Impact

Ask yourself a simple question. Which charities are doing a good job or even a great job? When you have settled on a few chosen organisations ask yourself what evidence you have for that. And when you think of organisations doing a great job do you have specific facts and figures in your head or just a general warm rosy glow about them?

In the numerous sessions with live audiences where I have asked these questions, the responses are always the same. People name a breadth of organisations, but when I press them on how they know that organisations are doing a good job, they tell me that they hear about them a lot in the media or that their newsletters had some really good facts and figures. And when I then ask for a specific fact or result that underpins their belief that a particular charity does a good job, just about nobody can give one.

So if we want people to support charities with their money, time, goodwill or energy then we have to help charities get better at measuring and communicating their impact.

Evidence of impact is our insurance policy against crisis of trust and confidence by the public. Evidence of impact is also the thing that reassures people that a donation is well spent.

## Challenge 4: The public knows very little about how modern charities work

## If there is a striking feature of our public awareness tracking for charities (we measure what the public think for around 50 charities) it is that the public have very little idea about how modern charities work. For example:

* The public overestimate how much is spent on fundraising and administration. They believe an acceptable amount to spend on admin would be around 10% of income and around 20% for fundraising. They think what actually happens is that around 35% of income is spent on administration and around 35% on fundraising.
* The public have little idea who is paid and who is unpaid in charities. While they think that volunteers are unpaid (phew!) they are more likely to think that trustees are paid than fundraisers. A substantial minority think that presidents are paid while far less think that patrons are paid.
* Over half think that £60k is too much to pay a CEO while nearly a quarter think that £20k or less is sufficient for a CEO to be paid.
* Few members of the public realise how big modern charities are, nor their sources of funding. Some of the biggest medical charities are very likely to be thought to have high levels of government funding – while both in fact receive next to no government funding.

6

The list could go on and on. Again and again we see the public perception of charities is seen through a rose-coloured fog of ignorance. The public trust charities but that trust is based not on evidence but on warm feelings.

**Challenge 5: Where can Charities do the best job in delivering public services?**

The debate about the delivery of public services by charities is a mixture of ideology interspersed with occasional nuggets of evidence. The reality is that those who advocate more public service delivery and those who argue against it rarely hold evidence-based policies. Indeed they appear to have decided what they think and then search for the evidence to support it. This is a shame.

Charities have a role to play in the delivery of public services but not all public services. We need a better understanding of where the charity ethos and structure can deliver better value than the commercial or public sector ethos, and where it cannot.

It does nobody any good to have public services delivered by anybody other than those who can do it most effectively. As Deng Xiao Ping loved to say “It does not matter if the cat is black or white as long as it catches the mouse”. We need to have a much better understanding of who can catch the mouse most effectively in public services.

**Challenge 6: How and When Do Infrastructure Investments Make a Difference?**

There has been substantial investment by the government and by the lottery bodies in developing the infrastructure for charities, social enterprises, community organisations and sports clubs. However the clarity about what works and what does not in developing the strengths of the sector is at best patchy. Government data from the citizenship survey shows that levels of volunteering have been flat for the last five years. This is despite hundreds of millions of pounds being invested in promoting volunteering.

If we want to develop the sector as a whole, we need to understand how we can make that investment in a way that will make a difference. One of the difficulties is that the bodies who are taking those funds are going to be the last ones who are likely to share any data that they may have on impact, unless it shows what they want it to show.

7

**CHAPTER 2**

**PROBLEM STATEMENT**

1. **Problem Statement:**

* Wish Lab is a web based application accessible for the employees of a company.
* This application helps the employees to fulfill the wishes of some needy.
* There are three main users : Admin, Volunteer and Employees.
* Admins can add the wishes list connected to the portal, assign volunteers for various tasks and can also view the details related to wishes.
* Employees are allowed to pick already added wishes, view already picked wishes, unpick any of the wishes and will have a provision to mark the wishes as placed.
* Volunteers will be assigned various tasks by the admin and will have options that help in fulfilling the assigned wishes.

9

**CHAPTER 3**

**PLANNING**

1. **Planning :**

We have planned the development of our web based application based on various user stories. The user stories are assigned on the basis of roles in the project.

There are basically 3 major roles in our web based application:

* Admin
* Employee
* Volunteer

For the **admin** we have planned to assign following tasks :

1. As a new admin of the WishLab application, he / she should be able to register to the application so that he / she can avail the features of the application as an admin.
2. As an Admin of the WishLab application, he / she should be able to login to the application and logout from it so that he / she can access all admin related functionalities securely.
3. As an Admin of the WishLab application, he / she should be able to change his / her password whenever required so that his / her account could be secured.
4. As an Admin of the WishLab application, he / she should be able to login even when he / she forgets his / her password using OTP by sending a mail to admin’s registered email Id.
5. As an Admin of the WishLab application, he / she should be able to view and update his / her profile so that he / she may remain updated till time.
6. As an Admin of the WishLab application, he / she should be able to add new wishes to the portal so that all the necessities could be satisfied.

11

1. As an Admin of the WishLab application, he / she should be able to view the picked wishes and view the status of wishes so that he / she would take required action.
2. As an Admin of the WishLab application, I should be able to assign volunteers to tasks so that there is proper communication with the employees.
3. As an Admin of the WishLab application, I should be able to grant credit points to the employee according to the picked wished so that the employees get motivated to fulfil more wishes.
4. As an Admin of the WishLab application, I should be able to assign credit points to the wishes on the basis of their priority so that each wish has its own value and advantage.
5. As an Admin of the WishLab application, I should be able to view employee details who has picked some wishes so that I can reach out to him/her.
6. As an Admin of the WishLab application, I should be able to prioritize the wishes, by adding an extra column to the database so that employee may know which wish needs to be fulfilled first.

12

For the **employee** we have planned to assign following tasks :

1. As an Employee of the WishLab application, I should be able to register to use the facility.
2. As an Employee of the WishLab application, I should be able to login to the application and logout from it so that I can access all the employee related functionalities securely.
3. As an Employee of the WishLab application , I should be able to change my password whenever required so that I can avail the features of the application as an employee.
4. As an Employee of the WishLab application , I should be able to login even when I forget my password using OTP so that I can avail features of WishLab.
5. As an Employee of the WishLab application, I should be able to view and update my profile so that I may remain updated till time.
6. As an Employee of the WishLab application, I should be able to view and pick the new wishes from the Wish List so that they can be fulfilled.

1. As an Employee of the WishLab application, I should be able to utilize the credit points earned from each wish so that I can maintain my enthusiasm in fulfilling the wish.

1. As an Employee of the WishLab application, I should be able to view the wishes picked along with its status so that I can know what wishes I have completed and which ones are pending.
2. As an Employee of the WishLab application, I should be able to deselect the wishes within a given period of time so that I can choose any other wish at any point of time.

13

1. As an Employee of the WishLab application, I should be able to mark the wishes as Submitted, once its submitted to volunteer so that I can remain updated.
2. As an Employee of the WishLab application, I should be able to view the volunteer details assigned to the task so that I can know which person is coming to visit me.

For the **volunteer** we have planned to assign following tasks :

1. As a Volunteer of the WishLab application, I should be able to register so that I can use the facility.
2. As a Volunteer of the WishLab application, I should be able to login to the application and logout from it so that I can access all volunteer related functionalities securely.
3. As a Volunteer of the WishLab application, I should be able to change my password whenever required so that I can avail the features of the application as an employee.
4. As a Volunteer of the WishLab application, I should be able to login even when I forget my password using OTP so that I can avail features of WishLab.
5. As a Volunteer of the WishLab application, I should be able to view and update my profile so that I may remain updated till time.
6. As a Volunteer of the WishLab application, I should be able to view the status of wishes so that I can know what changes needs to be done.

1. As a Volunteer of the WishLab application, I should be able to view the employee details from whose items are to be collected so that I can contact them.

14

1. As a Volunteer of the WishLab application, I should be able report the status regarding wishes to the admin so that admin can keep track of my progress.
2. As a Volunteer of the WishLab application, I should be able to accept or reject the request of admin so that I can tell if I am available at the time required or not.
3. As a Volunteer of the WishLab application, I should be able to upload photos on completion of the wishes so that progress of wishes could be shown.

15

**CHAPTER 4**

**DESIGN**

**4.Design**

We have used following technologies in the design of our web based application:

* Angular
* Typescript
* HTML
* CSS
* Bootstrap

* 1. **Angular :**

In Traditional web applications created using technologies such as Spring MVC and JSP, almost all the data (static HTML markup, dynamic model data, styling instruction, any other media etc.) comes directly from the server, having almost no client side processing.

This leads to issues like slow responses, more waiting time and almost no dynamic feedback while using these applications.

**What kind of applications can be built using Angular**?

* Angular is an open source framework for building both mobile and desktop web apps.
* Angular is exclusively used to build single page applications.
* Developers prefer TypeScript to write Angular code. But other than TypeScript, we can also write code using JavaScript (ES5 or ECMAScript 5)
* Angular provides a complete solution to develop client-side of an application and provides good support to all the server-side technologies like Java, .NET, etc.

**Why do most developers prefer TypeScript for Angular?**

* TypeScript is Microsoft’s extension for JavaScript which supports object oriented features and has strong typing system which enhances the productivity
* TypeScript supports many features like annotations, decorators, generics etc. A very good number of IDE’s like Sublime text, Visual Studio Code, Nodeclipse etc., are available with TypeScript support
* TypeScript code is compiled to JavaScript code using build tools like npm, bower, gulp, webpack etc., to make browser understand the code.

Angular is one of the most powerful client-side U.I. framework which can be used to develop **complex**, **customizable**, **modern**, **responsive** and **user friendly** web applications. Some such applications are PayPal, Netflix, Weather etc.

Angular is a single framework which addresses concernsof bothmobileanddesktop application.

From your existing knowledge of HTML, CSS, and JS, you would already know that to start writing anydynamic HTMLcode, we need the help of a **client-side**scripting language(like JS), where we can show specific data on HTML based on certain conditions.

But experienced U.I designers who have worked with JS frameworks, will point out some of the **flaws**with**JS** that we will soon see.

The Angular framework has chosen a **new statically-typed**, **client-side scripting** language called TypeScript, which beautifully works around most if not all such pitfalls of JavaScript. The Angular team recommends the usage of TypeScript for Angular applications.

Hence, to start building Angular applications, we would need to learn to write simple TypeScriptcode first and then look at the Angular application design.

**4.1.2 Features of Angular :**

Some of the features of Angular are as follows :

* **Easier to learn**: Angular is more modern and easier for developers to learn. It is a more streamlined framework where developers will be focusing on writing classes rather than the functional programming style of JavaScript.

* **Good IDE support**: Angular is written in TypeScript which is a superset of JavaScript and supports all ECMAScript 6 features. Many IDEs like Eclipse, Microsoft Visual Studio, Sublime Text etc., have good support for TypeScript.

* **Cross Platform**: Angular is a single platform which can be used to develop applications for multiple devices.

* **Lean and Fast**: Angular application's production bundle size (deployment size) is reduced by 100s of kilobytes due to which it loads faster during execution.

* **Simplicity**: Angular 4 has fewer number of directives as we use [ ] and ( ) for bindings in HTML elements.

* **Component-based**
  + Angular follows component based programming which is the future of web development. Each component we create is isolated from every other part of our application. This kind of programming allows us to use components written using other frameworks.
  + Inside a component, we write both business logic and view.
  + Every Angular application will have one top-level component and several sub components.

**4.1.3 Angular in Web Application Stack :**

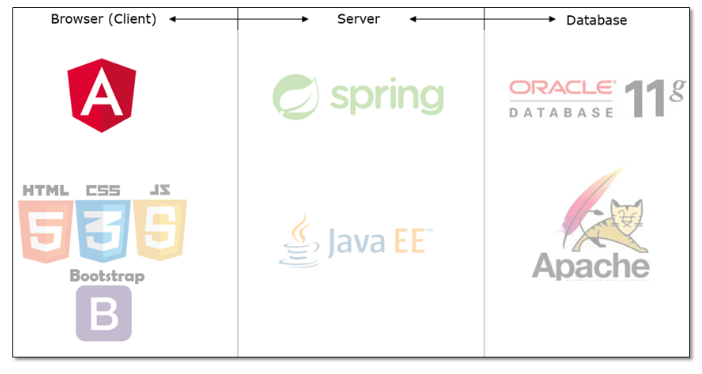
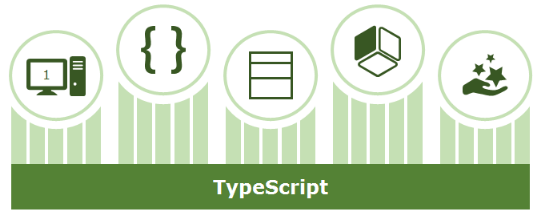


Figure 4.1.3 Angular in Web Application Stack

Angular places itself on the client side in the complete application stack and provides a complete client-side solution for swift application development. Angular has absolutely no dependencies and also gels perfectly with any possible server-side technology like Java, NodeJS, Spring, etc. and any database like Oracle hosted on Tomcat server can be used.

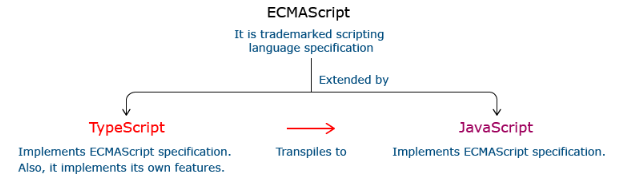
* 1. **Typescript :**

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

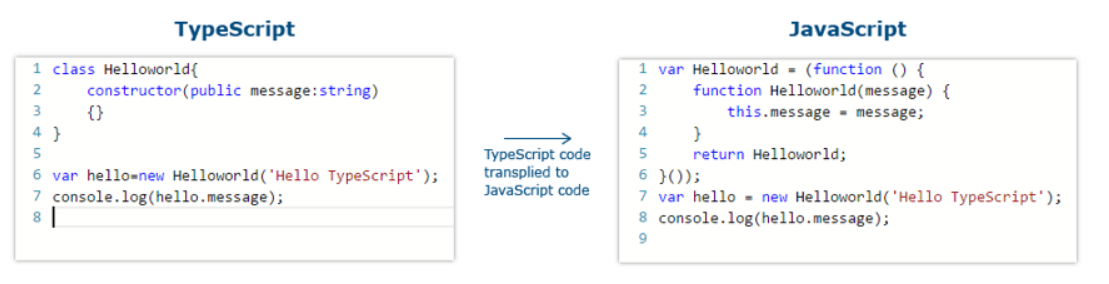


TypeScript is a typed **superset** of JavaScript that transpiles to JavaScript.

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



**4.2.1 Relationship between** **Typescript and Javascript :**



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

**4.2.2 Features of Typescript :**

Some of the features of Typescript are :

* **Static Typing:** It adds static typing to JavaScript, due to which the readability of the code improves and also helps in finding more early compilation errors than the runtime errors.
* **Modules support:** TypeScript provides an option to create modules so that we can modularize the code for easy maintenance. Modules also help in making the application scalable.
* **Object Oriented Programming:** TypeScript supports object oriented programming features such as classes, encapsulation, interface, inheritance and so on which help in creating highly structured and reusable code.
* **Open Source:** TypeScript is open source. The source code of TypeScript can be downloaded from github.
* **Cross Platform:** It works across platforms.
* **Tooling Support:** TypeScript works extremely well with Sublime Text, Eclipse, and almost all major IDEs as compared to JavaScript

**4.3 HTML**

**HTML**(**H**yper **T**ext **M**arkup **L**anguage)

* Hypertext is the text displayed on a computer which refers to other text or image that the user can access.
* Markup language is a set of tags which helps the web browser in presenting text in a document.



HTML can:

* Publish documents with text, headlines, images etc
* Create forms to collect user data.
* Include videos, audio clips, flash movies etc. inside an HTML document.
* Access online information via hyperlinks.

**4.3.1 Why we Need HTML ?**

Before World Wide Web, our applications used to look like this.

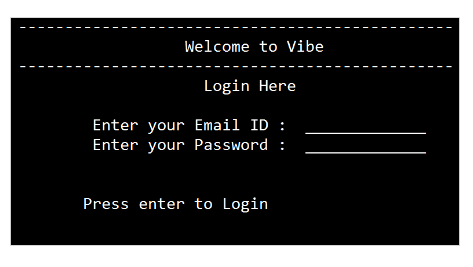
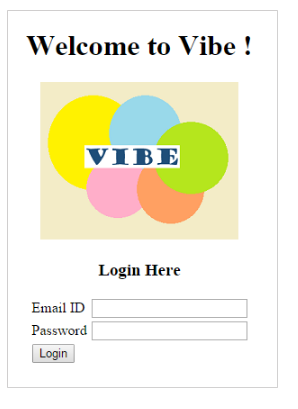


Figure 4.3.1 Web Application before World Wide Web

Due to its visual boredom, we started to design the web pages like this.



**4.3.3 Skeleton of a HTML Page:**

The basic structure of an HTML page can be understood by the figure shown below. It basically contains various tags such as <head>, <html>, <body>, <script> etc.

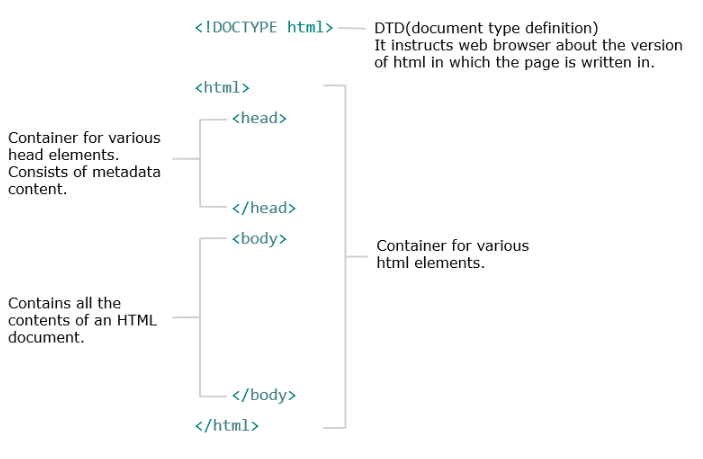


Figure 4.3.3 Skeleton of a HTML page

**4.3.4 Basic Syntax for an HTML Element:**

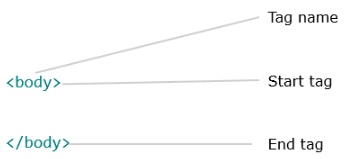


Figure 4.3.4 Basic Syntax for an HTML Element

**4.4 CSS (Cascading Style Sheet)**

****

CSS is a language that describes the style of an HTML document.CSS describes how HTML elements should be displayed.It is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.

It 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.

. **4.5 Bootstrap**

****

**Bootstrap** is HTML, CSS and JavaScript **framework** to quickly create **responsive webpages** by using its built-in components. It is a sleek, intuitive, and powerful mobile first front-end framework. It helps in quick development of responsive websites by providing HTML layout and CSS based templates for UI components like Forms, Tables, Navigation menus, Dropdowns, Carousel, etc. It uses HTML, CSS, and JS for faster and easier web development.

**Bootstrap:**

* was built at**Twitter Inc**. by Mark Otto and Jacob Thornton as a framework for internal tools
* is a **complete CSS framework** offering Grid system and configurations, Typography classes, UI components like forms, tables and more
* is a widely used framework in the development of **responsive**websites
* has a very **good documentation** which can be found in[Bootstrap site](http://getbootstrap.com/)

It helps in making responsive web pages by implementing **responsive web design** concept.

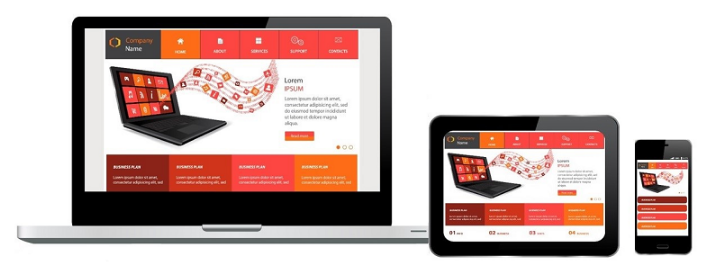


Figure 4.5.1 Responsive Web Design

Responsive web design is all about providing an **optimal viewing experience** across a wide variety of devices. This means that the entire content of the site should appear without loss of any information along with the maintenance of its appeal, when it is viewed in a mobile or a tablet or a desktop.

The basic features of responsive web design are as follows:

* **Fluid Layout** : Layout grows and shrinks based on the size of the device browser
* **Flexible Images** : Images adapt to the size of the device browser
* **Responsiveness** : Selectively apply CSS based on size of browser / device using media queries

**CHAPTER 5**

**CODING**

**CHAPTER 6**

**TESTING**