CHAPTER4

IMPLEMENTATION

This section explains about the different tool and technologies used in this project.

4.1 Resource requirement

Software requirement

Software used in the project are as follows

- Front End tools: HTML, JavaScript, CSS,
- Back End tools: python
- Browser that supports HTML and JavaScript
- IIS or apache server
- MySQL database
- Django

Hardware requirement

Hardware used in the project are as follows

- CPU: Pentium processor and above
- RAM: 2 GB & above
- HDD: 40 GB &above

4.1.1 Hypertext Markup Language

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

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

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

4.1.2 Cascading Style Sheets

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

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

4.1.3 Java script

JavaScript (sometimes abbreviated JS) is a prototype-based scripting language that is dynamic, weakly typed, general purpose programming language and has first-class functions. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

JavaScript was formalized in the ECMA Script language standard and is primarily used in the form of client-side JavaScript, implemented as part of a Web browser in order to provide enhanced user interfaces and dynamic websites. This enables programmatic access to computational objects within a host environment.

JavaScript's use in applications outside Web pages for example in PDF documents, site-specific browsers, and desktop widgets is also significant.

In this application, JavaScript is used for validation purpose like text box validation, email validation, phone number validation. JavaScript is the good tool for validating the web-applications.

4.1.4 Django - Python Framework

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java.The language provides constructs intended to enable writing clear programs on both a small and large scale.

Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source softwareand has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

4.1.5 IIS or apache server

Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Popular authentication modules include mod_access, mod_auth, mod_digest, and mod_auth_digest, the successor to mod_digest. A sample of other features include Secure Sockets Layer and Transport Layer Security support (mod_ssl), a proxy module (mod_proxy), a URL rewriting module (mod_rewrite), custom log files (mod_log_config), and filtering support (mod_include and mod_ext_filter).

Popular compression methods on Apache include the external extension

module, mod_gzip, implemented to help with reduction of the size (weight) of Web pages served over HTTP. ModSecurity is an open source intrusion detection and prevention engine for Web applications. Apache logs can be analyzed through a Web browser using free scripts, such as AWStats/W3Perl or Visitors.

Virtual hosting allows one Apache installation to serve many different Web sites. For example, one machine with one Apache installation could simultaneously serve www.example.com, www.example.org, test47.test-server.example.edu, etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs).

It supports password authentication and digital certificate authentication. Because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons

4.1.7 MySQL database

MySQL is a Relational Database Management System (RDBMS). MySQL server can manage many databases at the same time. In fact, many people might have different databases managed by a single MySQL server. Each database consists of a structure to hold the data and the data itself. A data-base can exist without data, only a structure, be totally empty, twiddling its thumbs and waiting for data to be stored in it.

Data in a database is stored in one or more tables. You must create the data-base and the tables before you can add any data to the database. First you create the empty database. Then you add empty tables to the database. Database tables are organized like other tables that you're used in rows and columns. Each row represents an entity in the database, such as a customer, a book, or a project. Each column contains an item of information about the entity, such as a customer name, a book name, or a project start date. The place where a particular row and column intersect, the individual cell of the table, is called a field. Tables in databases can be related. Often a row in one table is related to several rows in another table. For instance, you might have a database containing data about books you own. You would have a book table and an author table. One row in the author table might contain information about the author of several books in the book table. When tables are related, you include a column in one table to hold data that matches data in the column of another table

4.1.8 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL is a relational database management system.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. The MySQL Database Server is very fast, reliable, and easy to use.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

MySQL Server works in client/server or embedded systems.
The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

4.2 Discussion of code segment

In the following segment, we are going to discuss about the code segment.

4.2.1 Forms code

```
from django import forms
from authentication.models import UserProfileInfo
from django.contrib.auth.models import User
    class UserForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput())
    class Meta():
        model = User
        fields = ('username','password','email')
class UserProfileInfoForm(forms.ModelForm):
 class Meta():
     model = UserProfileInfo
```

```
        fields = ('portfolio_site','profile_pic')
```

database generatio code

4.2.2 Models code

```
from django.db import models
from django.contrib.auth.models import User
class UserProfileInfo(models.Model):
    user = models.OneToOneField(User,on_delete=models.CASCADE)
    portfolio_site = models.URLField(blank=True)
    profile_pic = models.ImageField(upload_to='profile_pics',blank=True)
    def __str__(self):
      return self.user.username
def create_profile(sender, **kwargs):
    if kwargs['created']:
        user_profile = UserProfileInfo.objects.create(user=kwargs['instance'])
post_save.connect(create_profile, sender=User)
class Companies(models.Model):
    name=models.CharField(max_length=20,blank=False)
    based_on=models.CharField(max_length=20,blank=True)
    founder=models.CharField(max_length=20,blank=False)
    net_worth=models.IntegerField(blank=True)
    yr_of_estb=models.DateField()
class Employee(models.Model):
    c_id=models.ForeignKey(Companies,on_delete=models.CASCADE)
    name=models.CharField(max_length=20,blank=False)
    address=models.TextField(blank=False)
    age=models.IntegerField()
    email=models.EmailField()
    ph_no=models.IntegerField(blank=False)
    def __str__(self):
        return self.name
class Branches(models.Model):
    c_id=models.ForeignKey(Companies,on_delete=models.CASCADE)
    mang_id=models.ForeignKey(Employee,on_delete=models.CASCADE)
    no_of_employee=models.IntegerField(blank=False)
    location=models.CharField(max_length=20,blank=False)
    shift=models.CharField(max_length=10,blank=False)
    def __str__(self):
        return str(self.id)
class Placement_dept(models.Model):
    c_id=models.ForeignKey(Companies,on_delete=models.CASCADE)
    hr_id=models.ForeignKey(Employee,on_delete=models.CASCADE)
    no_of_dept=models.IntegerField()
    req_emp=models.IntegerField()
    year=models.DateField()
class Oncampus(models.Model):
    c_id=models.ForeignKey(Companies,on_delete=models.CASCADE)
    p_id=models.ForeignKey(Placement_dept,on_delete=models.CASCADE)
    campus_name=models.CharField(max_length=20,blank=False)
    no_of_sel=models.IntegerField(blank=False)
    no_of_rej=models.IntegerField(blank=False)
    location=models.CharField(max_length=20,blank=False)
    avg_salary=models.IntegerField(blank=False)
class Offcampus(models.Model):
    p_id=models.ForeignKey(Placement_dept,on_delete=models.CASCADE)
    no_of_stu_rec=models.IntegerField()
    region=models.CharField(max_length=20,blank=True)
    avg_salary=models.IntegerField(blank=False)
class Annual_report(models.Model):
    c_id=models.ForeignKey(Companies,on_delete=models.CASCADE)
    profit=models.IntegerField()
    anu_invst=models.IntegerField()
    loss=models.IntegerField()
    year=models.DateField()
```

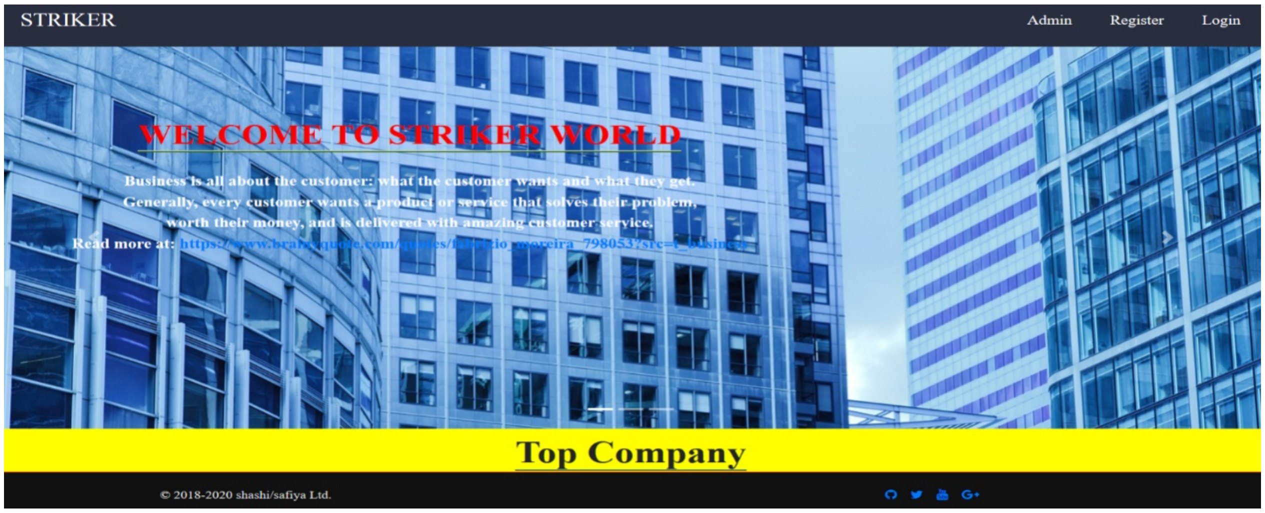
4.2.3 Views code

```
from django.shortcuts import render
from authentication.forms import UserForm,UserProfileInfoForm
from django.contrib.auth import authenticate, login, logout
from django.http import HttpResponseRedirect, HttpResponse
from django.urls import reverse
from django.contrib.auth.decorators import login_required
from authentication.models import *
from django.db import connection
from django.db.models import Q
def index(request):
    return render(request,'authentication/index.html')
@login_required
def special(request):
    return HttpResponse("You are logged in !")
@login_required
def user_logout(request):
    logout(request)
    return HttpResponseRedirect(reverse('index'))
def register(request):
    registered = False
    if request.method == 'POST':
        user_form = UserForm(data=request.POST)
        profile_form = UserProfileInfoForm(data=request.POST)
        if user_form.is_valid() and profile_form.is_valid():
            user = user_form.save()
            user.set_password(user.password)
            user.save()
            profile = profile_form.save(commit=False)
            profile.user = user
            if 'profile_pic' in request.FILES:
                print('found it')
                profile.profile_pic = request.FILES['profile_pic']
            profile.save()
            registered = True
        else:
            print(user_form.errors,profile_form.errors)
    else:
        user_form = UserForm()
        profile_form = UserProfileInfoForm()
    return render(request,'authentication/registration.html',
                          {'user_form':user_form,
                           'profile_form':profile_form,
                           'registered':registered})
```

```
def user_login(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(username=username, password=password)
        if user:
            if user.is_active:
                login(request,user)
                return HttpResponseRedirect(reverse('index'))
            else:
                return HttpResponse("Your account was inactive.")
        else:
            print("Someone tried to login and failed.")
            print("They used username: {} and password: {}".format(username,password))
            return HttpResponse("Invalid login details given")
    else:
        return render(request, 'authentication/login.html', {})
def Companyn(request):
    cmpny_list=Companies.objects.all()
    context={
    'cmpny_list':cmpny_list }
    print(cmpny_list)
    return render(request,'authentication/home.html',context)
def Details(request,id):
     context={
    'a':a,
     'c':id, }
    return render(request,'authentication/details.html',context)
def Annual(request,c):
    p=Companies.objects.get(pk=c)
    context={
    'p':p}
    return render(request,'authentication/annual.html',context)
def Mandetail(request,d):
    row=Employee.objects.all()
    return render(request,'authentication/manager.html',context)
def Pladetails(request,j):
    c=Companies.objects.get(pk=j)
    context={
        'c':c }
    print(c)
    return render(request,'authentication/placement.html',context)
def Search(request):
    if request.method=='POST':
        srch=request.POST['srh']
        if srch:
            match=Employee.objects.filter(Q(name__icontains=srch))
            if match:
                return render(request,'authentication/index.html',{'sr':match})
            else:
                message.error(request,'no result found')
        else:
            return HttpResponseRedirect('/search/')
    return render(request,'authentication/index.html')
```

4.3 Discussion of the results

Home page



WELCOME TO STRIKER WORLD

Business is all about the customer: what the customer wants and what they get. Generally, every customer wants a product or service that solves their problem, worth their money, and is delivered with amazing customer service.
Read more at: https://www.brainyquote.com/quotes/fabrizio_moreira_798053?src=t_business

Top Company

© 2018-2020 shashi/safiya Ltd.

Fig 4.1 Home Page

As shown in the above fig4.1the login page that gives the user the option to register ,log in or register are displayed on this page.

Login page

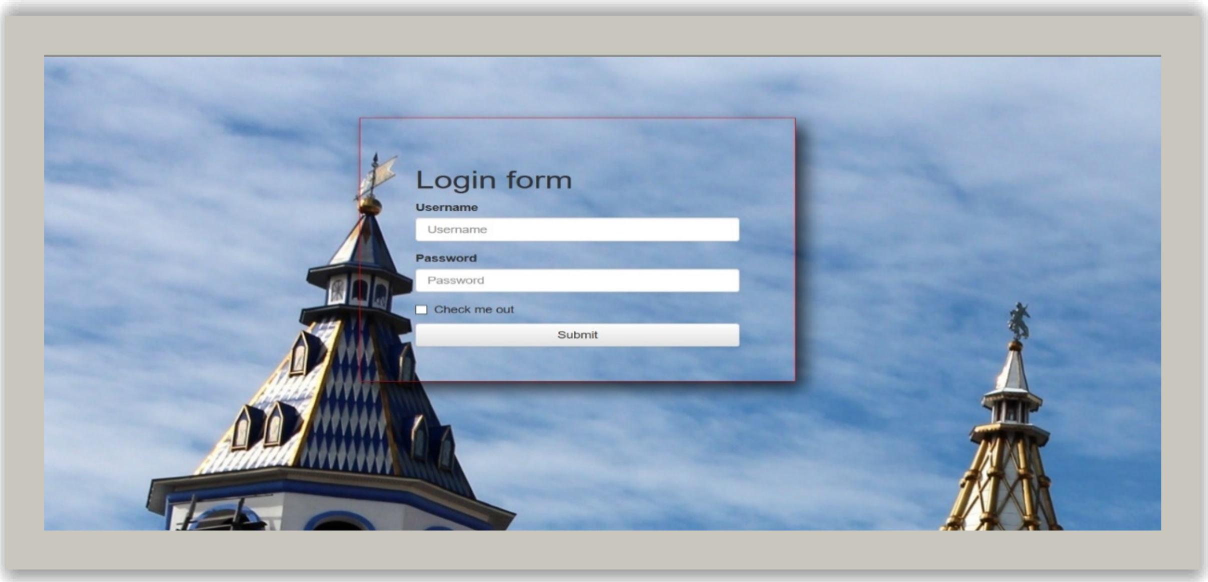
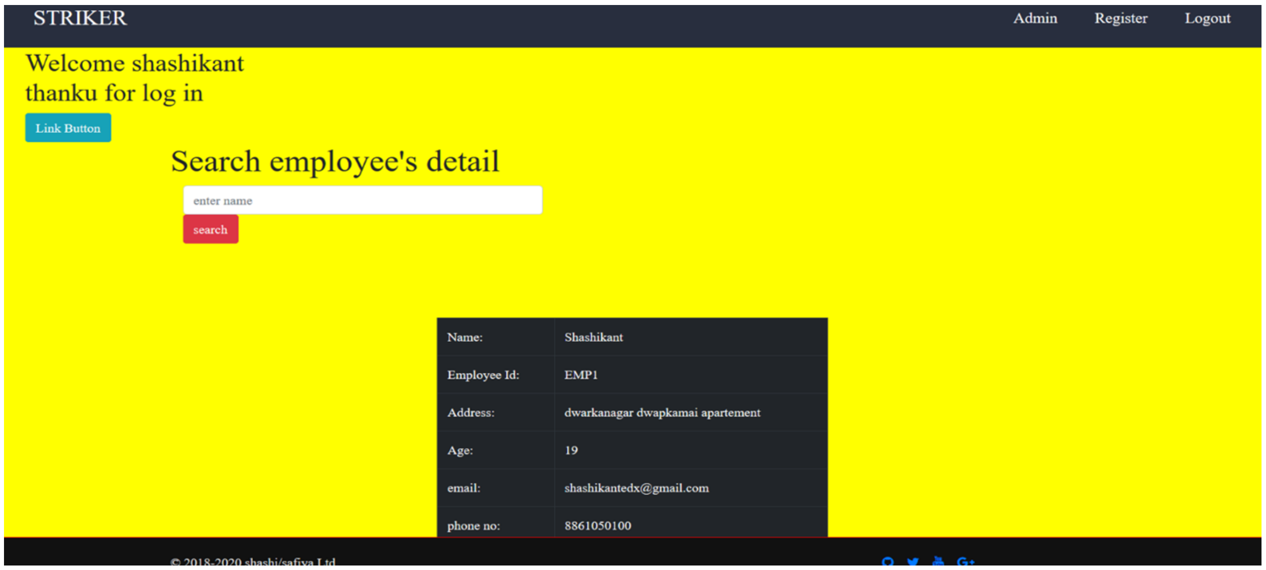


Fig 4.2 Login Page

As shown in the above fig 4.2 the login page of the website is shown in this page.This page you to the home page when the right username and password is entered.

Search page



Welcome shashikant
thanku for log in

Search employee's detail

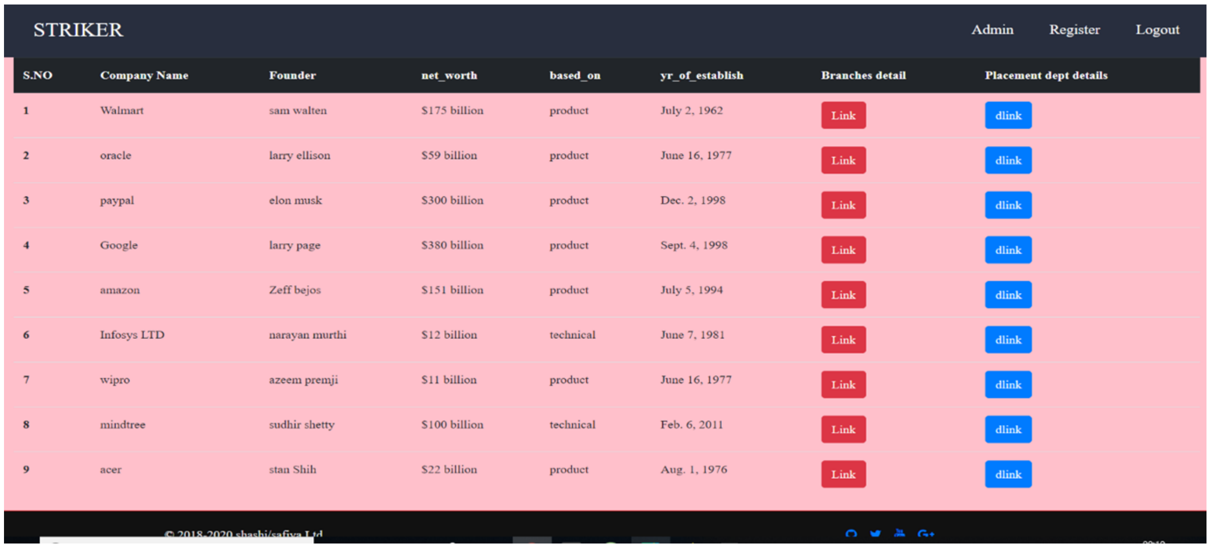
Name:	Shashikant
Employee Id:	EMP1
Address:	dwarkanagar dwapkamai apartement
Age:	19
email:	shashikantedx@gmail.com
phone no:	8861050100

© 2018-2020 shashi/safiya Ltd

Fig 4.3 Search Page

As shown in the above fig 4.3 the search page of the website that lets the user to search the details of the employees on this page.

Detail page



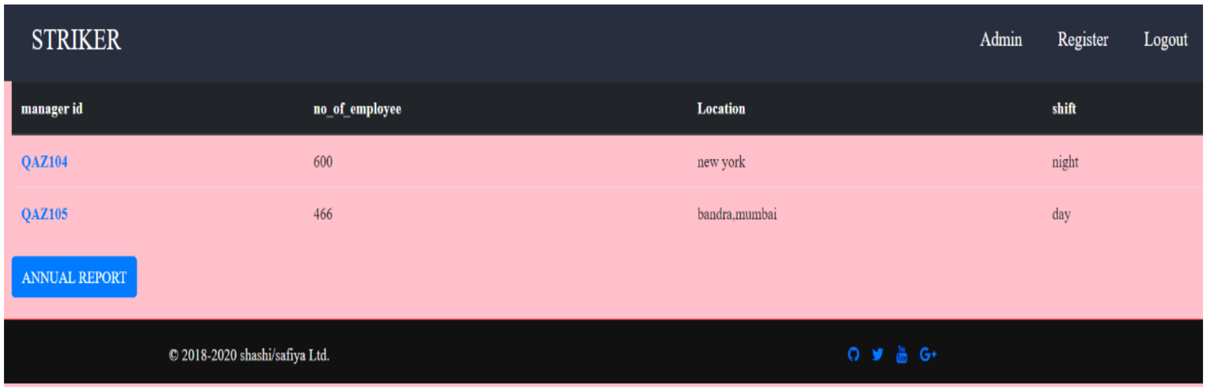
S.NO	Company Name	Founder	net_worth	based_on	yr_of_establish	Branches detail	Placement dept details
1	Walmart	sam walten	$175 billion	product	July 2, 1962	Link	dlink
2	oracle	larry ellison	$59 billion	product	June 16, 1977	Link	dlink
3	paypal	elon musk	$300 billion	product	Dec. 2, 1998	Link	dlink
4	Google	larry page	$380 billion	product	Sept. 4, 1998	Link	dlink
5	amazon	Zeff bejos	$151 billion	product	July 5, 1994	Link	dlink
6	Infosys LTD	narayan murthi	$12 billion	technical	June 7, 1981	Link	dlink
7	wipro	azeem premji	$11 billion	product	June 16, 1977	Link	dlink
8	mindtree	sudhir shetty	$100 billion	technical	Feb. 6, 2011	Link	dlink
9	acer	stan Shih	$22 billion	product	Aug. 1, 1976	Link	dlink

© 2018-2020 shashi/safiya Ltd

Fig 4.4 Detail Page

As shown in the above fig 4.4 shows the detail page, which contains the collection of various companies

Branch detail page

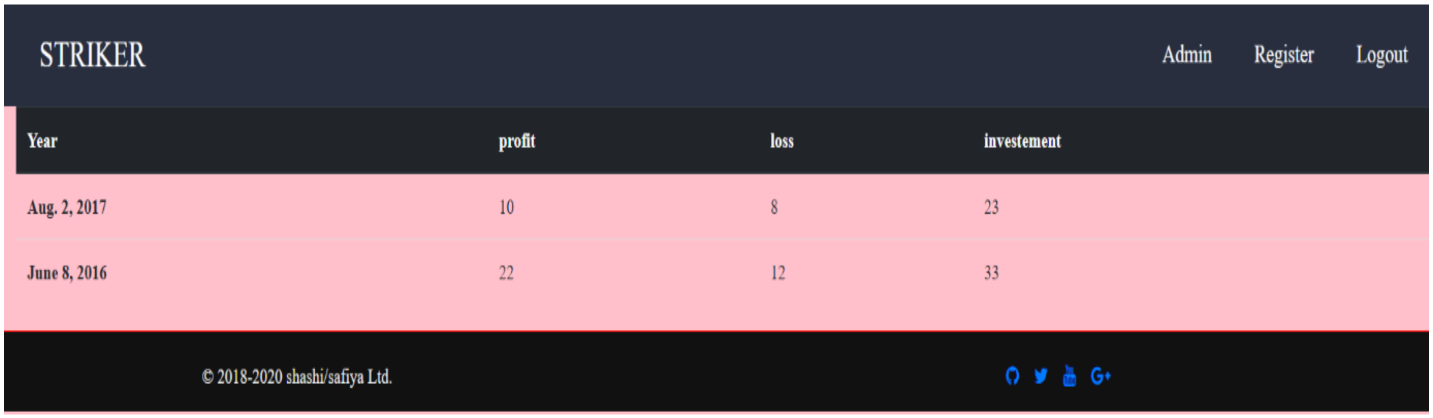


manager id	no_of_employee	Location	shift
QAZ104	600	new york	night
QAZ105	466	bandra,mumbai	day

Fig 4.5 Branch Detail Page

As shown in the above fig 4.5 the branch detail of the company that are the manager id, number of employees, location and the shifts of the employees are displayed in this page.

Annual report page

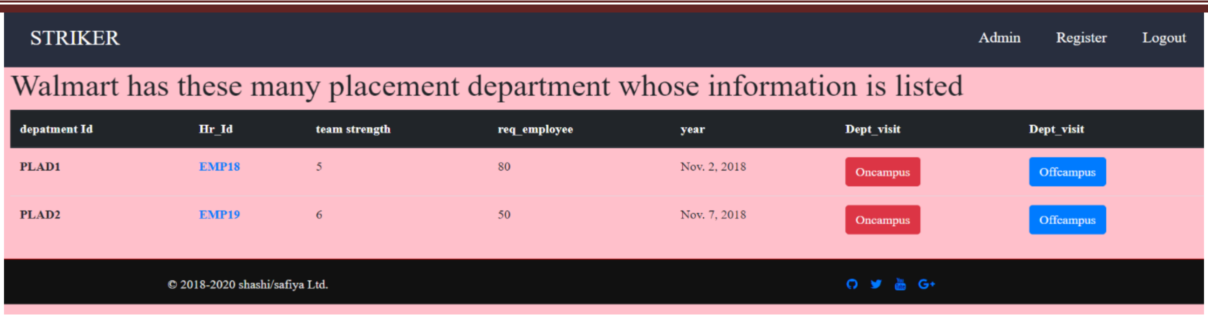


Year	profit	loss	investement
Aug. 2, 2017	10	8	23
June 8, 2016	22	12	33

Fig 4.6 Annual Report Page

As shown in the above fig 4.6 the annual report of company like the profit and the loss of the company are displayed in the page.

Placement department detail page



Walmart has these many placement department whose information is listed

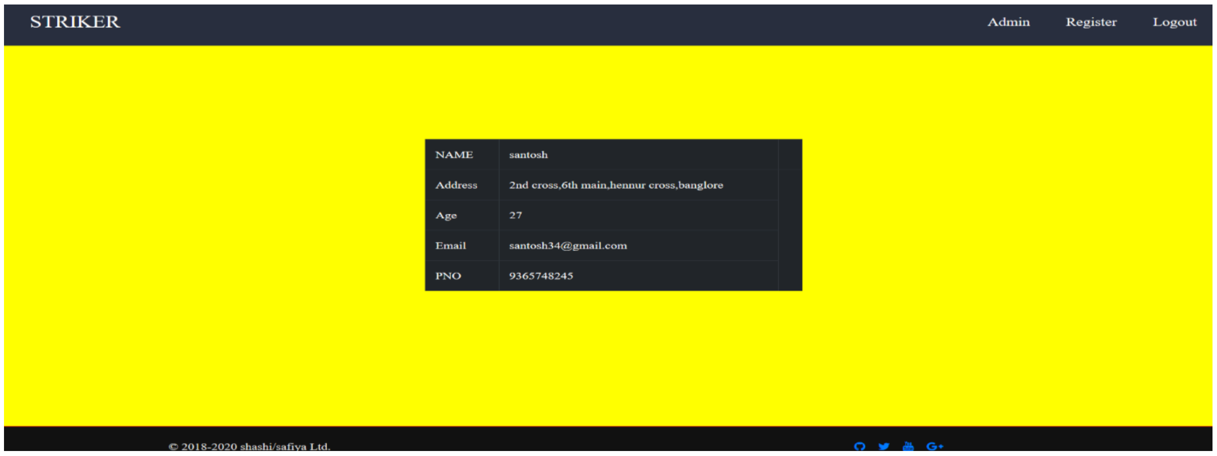
depatment Id	Hr_Id	team strength	req_employee	year	Dept_visit	Dept_visit
PLAD1	EMP18	5	80	Nov. 2, 2018	Oncampus	Offcampus
PLAD2	EMP19	6	50	Nov. 7, 2018	Oncampus	Offcampus

© 2018-2020 shashi/safiya Ltd.

Fig 4.7 Placement department detail page

As shown in the above fig 4.7 the detail of placement department displayed on this page.

Manager detail page



NAME	santosh
Address	2nd cross,6th main,hennur cross,banglore
Age	27
Email	santosh34@gmail.com
PNO	9365748245

© 2018-2020 shashi/safiya Ltd.

4.8 Manager detail page

As shown in the above fig 4.8 the manager details like the name of the manager, address, age, email, phone number are displayed on this page.

4.4 Applications

- This project helps for clients to get the list of emerging companies within seconds.
- They doesn't need to roam around streets of cities or contact any people for search of companies
- Once they login through this website they get the list of companies.
- They can further click on the company for more details about the branches and their managers.
- They can also get the annual report of the company.
- They can get the information about each employee even his mode of placement in the company
- Hence, the overall process has become easy.

4.5 Discussion of the results

- The given project is able to store the information of the different companies present
- The client can login and check for list of companies
- The client can also get the information about various branches and their details
- The client can see the annual report of the company