

PROFESSIONAL TRAINING REPORT
AT
Sathyabama Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in
Computer Science and Engineering

By

SURYA S
(Reg No: 39120123)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119,
TAMILNADU.

NOVEMBER 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SURYA.S**
(Reg. No. : **391120123**) who carried out the project entitled “**JOURNAL
PUBLICATION ONLINE**”

Internal Guide
Dr.S.GOWRI, M.E., Ph.D.,

Head of the Department
Dr.R.SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, SURYA S (Reg. No. 39120123) hereby declare that the Professional Training Report on “**WEB APPLICATION DEVELOPMENT WITH ANGULAR.JS ON Journal publication online** ” done by me under the guidance of **Dr.R.SUBHASHINI, M.E., Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Thechnology degree in informaton technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to, **Dr. T. SASIKALA M.E., Ph.D.**, Dean, School of Computing and **Dr.R.SUBHASHINI, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.GOWRI, M.E., Ph.D.**, for her valuable guidance, suggestions which paved way for the successful completion of my project work.

I wish to express my thanks to all teaching and non-teaching staff members of the department of Information Technology who were helpful in many ways for the completion of project.

TRAINING CERTIFICATE



ABSTRACT

In the Journal publication application website the blogs are published. The published journal are done by the user registered to our website and the user and the journal are verified by the admin. The user registered details are stored in our mongodb server. The journal has the following details like title, author, description, content. User can be able to update their journal whenever they want. Admin can able to review the published journals and application can be accept or reject it. The published journals were also stored in our mongodb server. Users can see the others published journals but it can not be edit and delete it. It can be done by verifying that the user has the rights to perform the operation by using authorization in our nodejs. Users and admins can search the journals by the journals name to view it. For the admin login alone the manage post is visible is in header. When admin click the manage post button it redirect to manage post and we can see the post details of the user. When the user add a post that is visible to admin in manage post and it will visible in home page.

TABLE OF CONTENTS		
CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	5
1	INTRODUCTION	
	1.1 About The Concept	8
2	Aim And Scope Of The Present Investigation	
	2.1 Aim	9
	2.2 Scope	9
	2.3 Project Flow	10
	2.3.1 Main Page	10
	2.3.2 Login Page	10
	2.3.3 Login Page Working Flow	11
	2.3.4 Register Page Working Flow	12
	2.3.5 After Login Page (USER PAGE)	12
	2.3.6 After Login Page (ADMIN PAGE)	13
3	Experimental (OR) Materials And Methods, Algorithm Used	
	3.1 Software Required	14
	3.1.1 Windows 10	14
	3.1.2 Node JS	14
	3.1.3 Angular CLI	16
	3.2 Hardware Required	
	3.2.1 Hard Disk	18
	3.2.2 RAM	18
	3.2.3 Processor	18
	3.2.4 Mouse	19
	3.2.5 Keyboard	19
	3.2.6 Speaker	19
4	Testing	
	4.1 System Testing	21
	4.2 Unit Testing	21
	4.3 Integration Testing	22
5	Result And Discussion, Performance Recommendation	
	5.1 Result And Discussion	23
	5.2 Recommendation	23
	5.3 Screen Shots	24
	5.4 Summary And Conclusion	

6	References	55
6.1 Appendix		24
6.2 Sample Screen		57

CHAPTER - 1

INTRODUCTION

1.1 ABOUT THE CONCEPT

A journal Publication is a **scholarly publication containing articles written by researchers, professors and other experts**. Journals focus on a specific discipline or field of study. Unlike newspapers and magazines, journals are intended for an academic or technical audience, not general readers.

Journal publication helps **to preserve your work in the permanent records of research in the field**. Adding your work to this record involves you in the active research community for a topic, helping to expand your professional network, increasing potential for collaboration and interaction with peers.

CHAPTER - 2

AIM AND SCOPE OF PRESENT INVESTIGATION

2.1 AIM

Journals Publications serve a range of purposes. A journal is meant **collect your ideas and observations on any number of things and put the happenings of each day into writing**. In this way, you are able to better remember what you did, what you thought, and what was happening when you were younger.

It builds corporate enthusiasm and supports ongoing education. When you share your published articles internally, it makes others feel proud that they are part of an known and respected organization. Additionally, sharing published pieces with employees helps them better understand marketplace challenges and solutions.

2.2 SCOPE

Publishing journals of research projects in peer-reviewed journals **enables the scientific and medical community to evaluate the findings themselves**. It also provides instructions so that other researchers can repeat the experiment or build on it to verify and confirm the results.

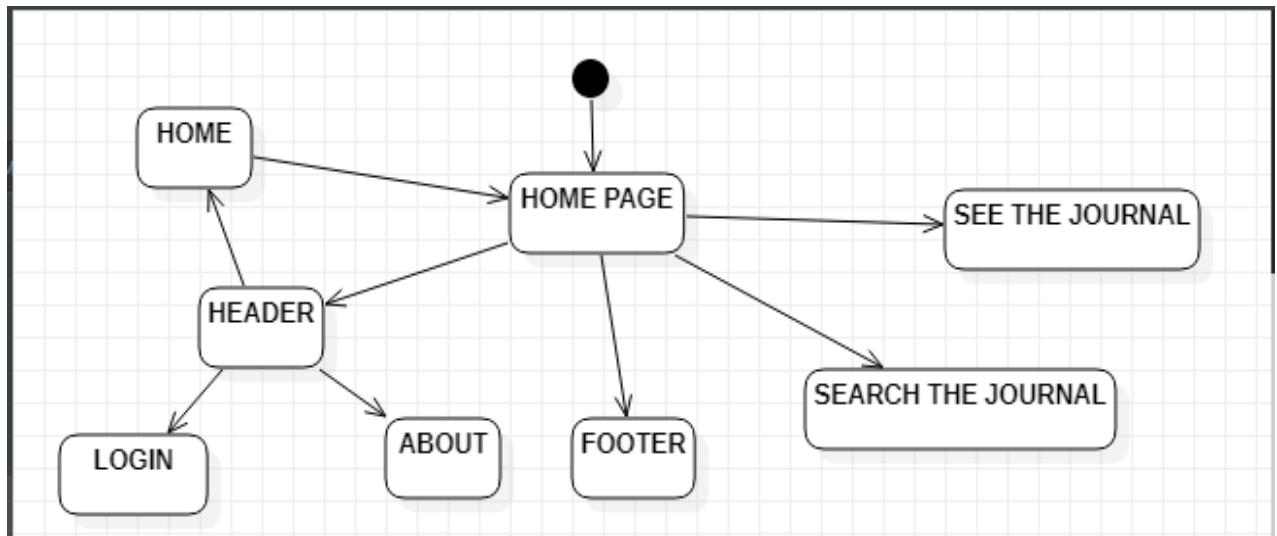
Journal publications include studies/articles/papers which are peer-reviewed. This means that any article included in a journal, has been very rigorously reviewed against strict criteria and standards by a panel of researchers or scholars in the field.

- Journal publications are a rich source of information: they aim to continually challenge and build new knowledge about your subject area.
- Your module handbook will identify a selection of journal publications which are useful and directly relevant to your subject.

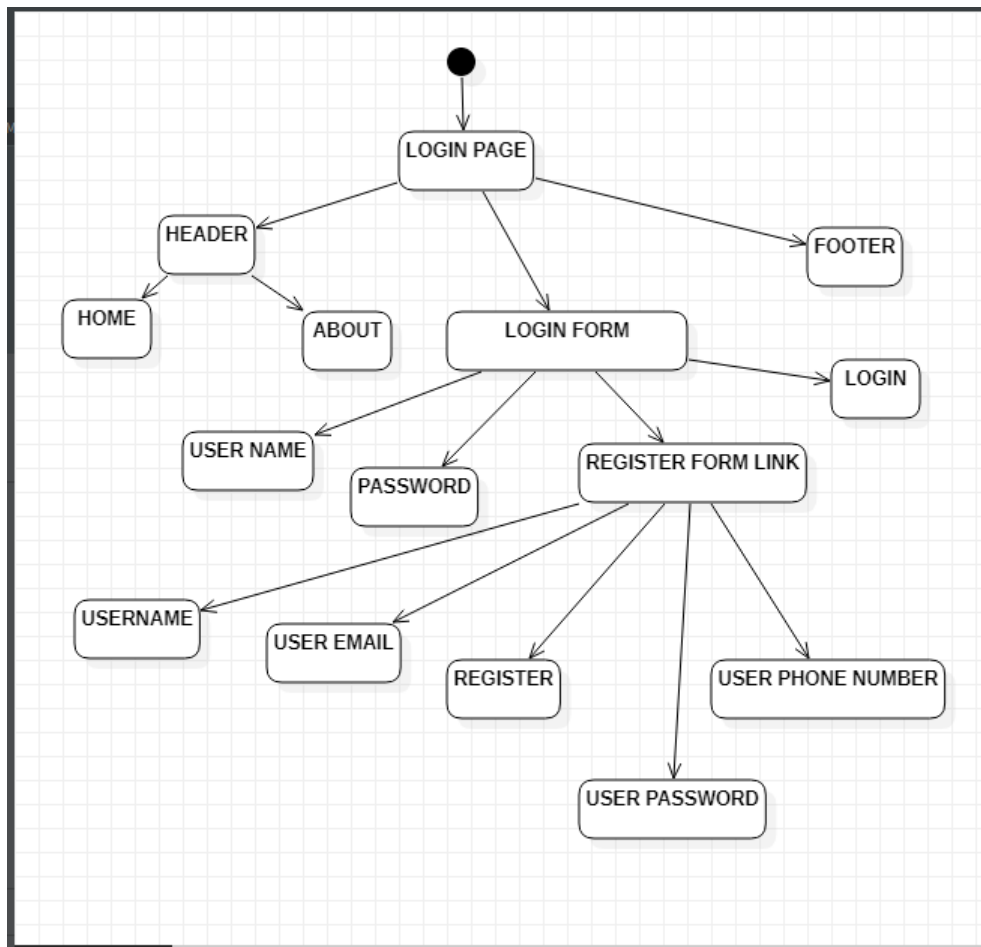
In your assignments it is expected that you will refer to a range of research articles to develop your discussion of the particular topic.

2.3 PROJECT FLOW

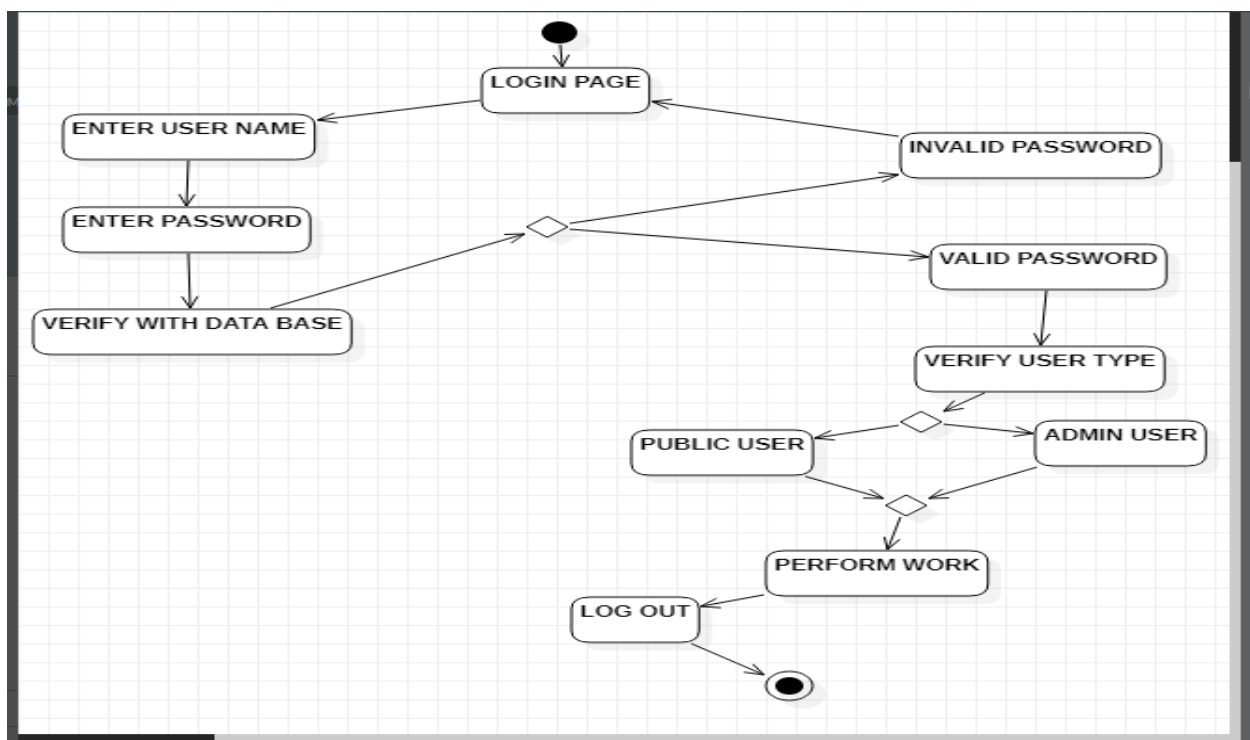
2.3.1 MAIN PAGE



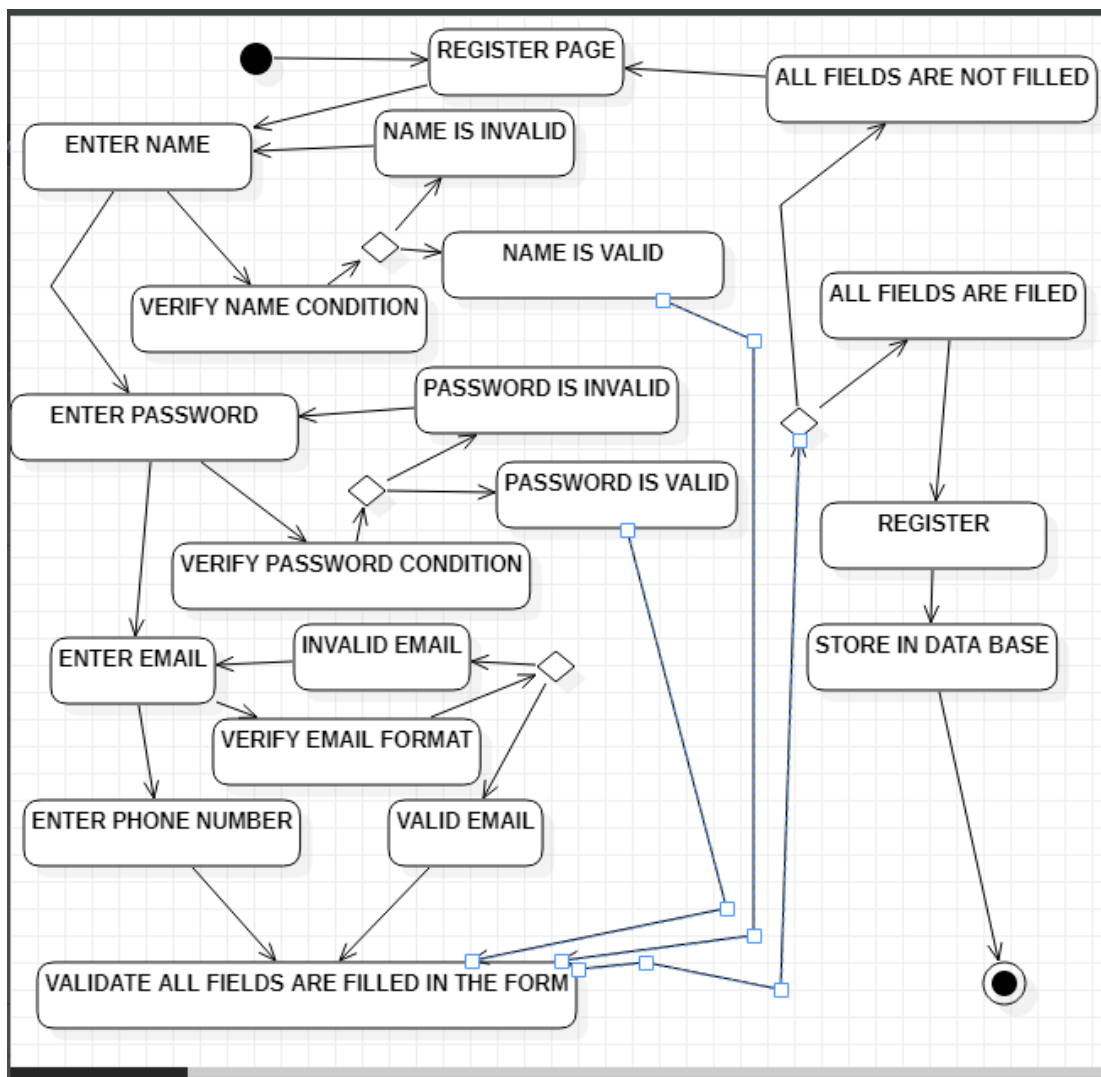
2.3.2 LOGIN PAGE



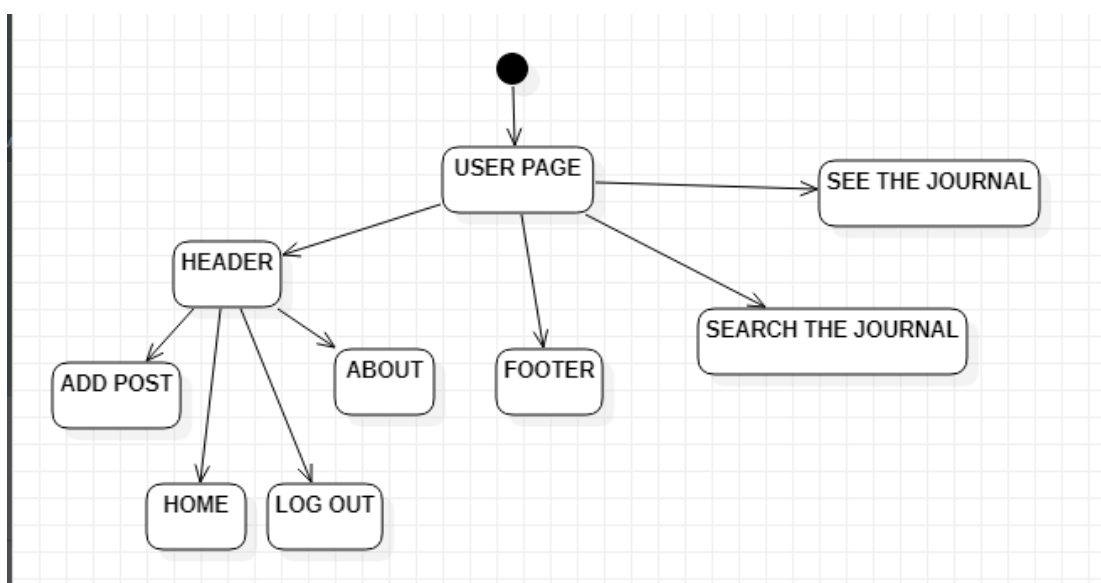
2.3.3 LOGIN PAGE WORKING FLOW



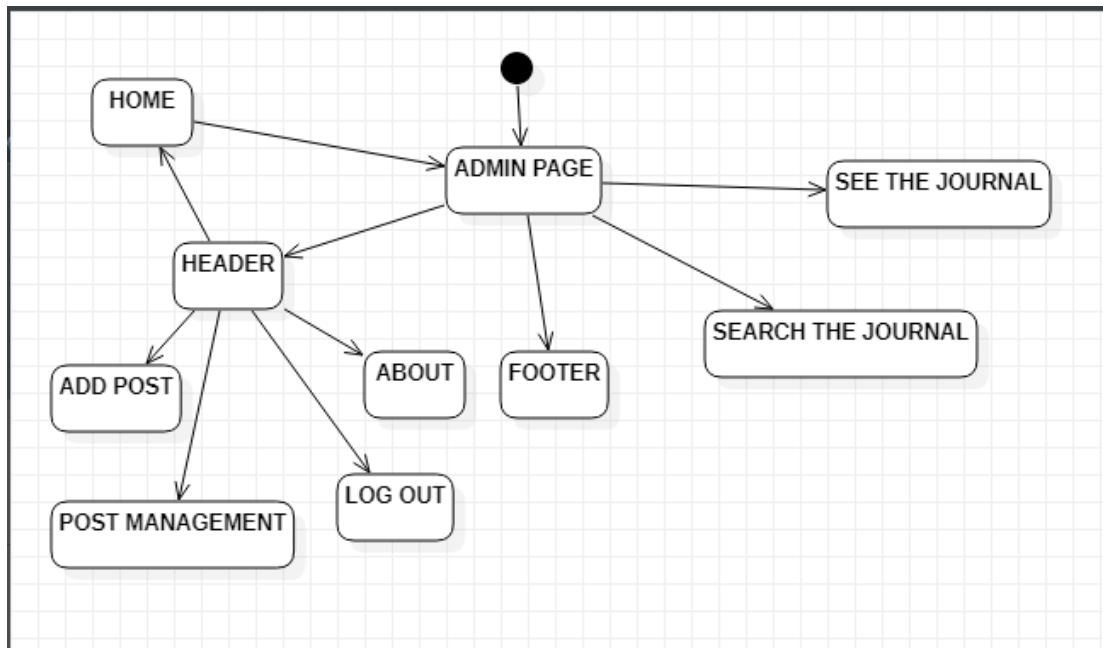
2.3.4 REGISTER PAGE WORKING FLOW



2.3.5 AFTER LOGIN PAGE (USER PAGE)



2.3.6 AFTER LOGIN PAGE (ADMIN PAGE)



CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS

3.1 SOFTWARE REQUIREMENT

- Windows 10 or above
- Node JS
- Angular CLI
- Database

3.1.1 Windows 10 or above

Windows 10 is a Microsoft operating system for personal computers, tablets, embedded devices and internet of things devices.

Microsoft released Windows 10 in July 2015 as a follow-up to Windows 8. The company has said it will update Windows 10 in perpetuity rather than release a new, full-fledged operating system as a successor.

3.1.2 Node JS

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

This is in contrast to today's more common concurrency model, in which OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use. Furthermore, users of Node.js are free from worries of dead-locking the process, since there are no locks. Almost no function in Node.js directly performs I/O, so the process never blocks except when the I/O is performed using synchronous methods of Node.js standard library. Because nothing blocks, scalable systems are very reasonable to develop in Node.js.

If some of this language is unfamiliar, there is a full article on Blocking vs. Non-Blocking.

Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. Typically, behavior is defined through callbacks at the beginning of a script, and at the end a server is started through a blocking call like **EventMachine::run()** In Node.js, there is no such start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no more callbacks to perform. This behavior is like browser JavaScript — the event loop is hidden from the user.

HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js being designed without threads doesn't mean you can't take advantage of multiple cores in your environment. Child processes can be spawned by using our **child_process.fork()** API, and are designed to be easy to communicate with. Built upon that same interface is the **cluster** module, which allows you to share sockets between processes to enable load balancing over your cores.



3.1.3 Angular CLI

This topic can help you understand Angular: what Angular is, what advantages it provides, and what you might expect as you start to build your applications.

Angular is a development platform, built on TypeScript. As a platform, Angular includes:

A component-based framework for building scalable web applications

A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more

A suite of developer tools to help you develop, build, test, and update your code

With Angular, you're taking advantage of a platform that can scale from single-developer projects to enterprise-level applications. Angular is designed to make updating as straightforward as possible, so take advantage of the latest developments with a minimum of effort. Best of all, the Angular ecosystem consists of a diverse group of over 1.7 million developers, library authors, and content creators.

This section explains the core ideas behind Angular. Understanding these ideas can help you design and build your applications more effectively.

- **Components**

Components are the building blocks that compose an application. A component includes a TypeScript class with a **@Component()** decorator, an HTML template, and styles. The **@Component()** decorator specifies the following Angular-specific information:

A CSS selector that defines how the component is used in a template. HTML elements in your template that match this selector become instances of the component.

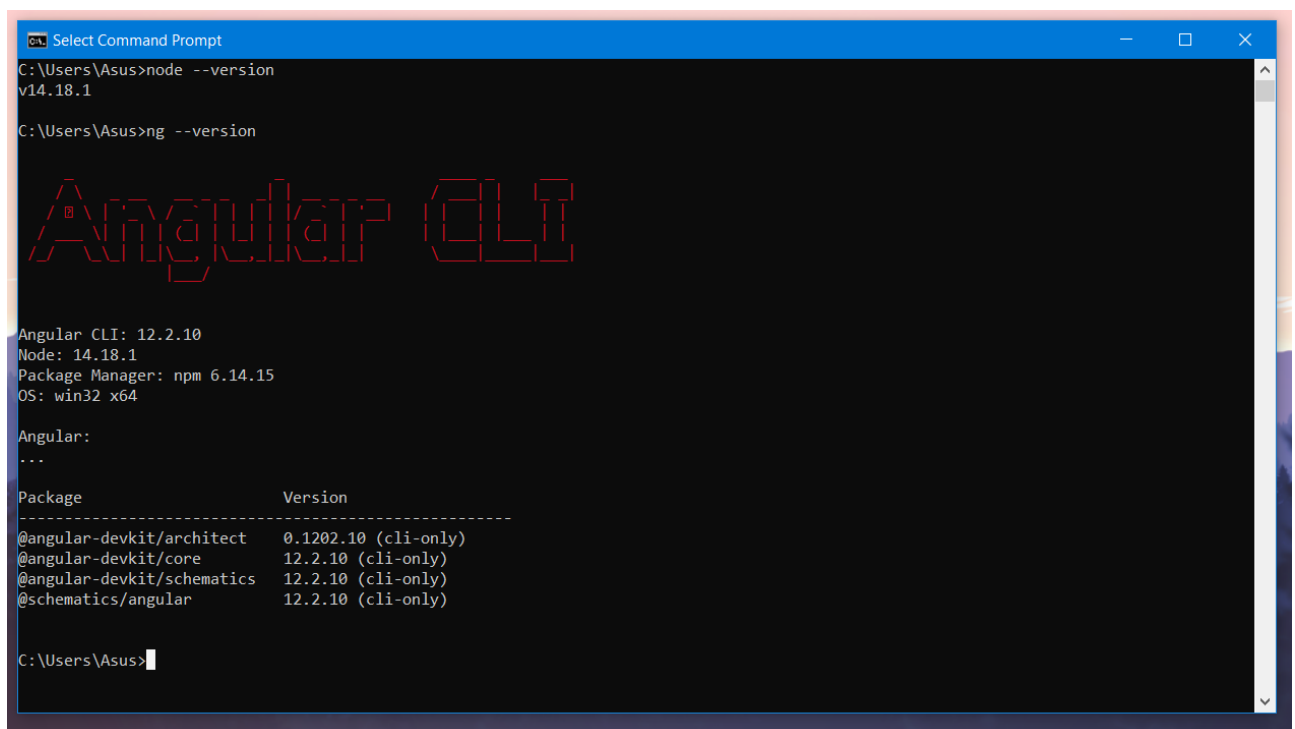
An HTML template that instructs Angular how to render the component.

An optional set of CSS styles that define the appearance of the template's HTML elements.

- **Templates**

Every component has an HTML template that declares how that component renders. You define this template either inline or by file path.

Angular extends HTML with additional syntax that lets you insert dynamic values from your component. Angular automatically updates the rendered DOM when your component's state changes. One application of this feature is inserting dynamic text, as shown in the following example.



```

C:\Users\Asus>node --version
v14.18.1

C:\Users\Asus>ng --version

Angular CLI: 12.2.10
Node: 14.18.1
Package Manager: npm 6.14.15
OS: win32 x64

Angular:
...
Package          Version
-----
@angular-devkit/architect    0.1202.10 (cli-only)
@angular-devkit/core        12.2.10 (cli-only)
@angular-devkit/schematics   12.2.10 (cli-only)
@schematics/angular         12.2.10 (cli-only)

C:\Users\Asus>
```

3.2 HARDWARE REQUIREMENT

- Hard Disk
- RAM – 1 GB
- Processor – Dual Core or Above
- Mouse
- Keyboard
- Monitor

3.2.1 Hard Disk

A hard disk or fixed disk, is an electro-mechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.

During the mid-1990s the typical hard disk drive for a PC had a capacity of about 1 gigabyte. As of January 2019, desktop hard disk drives typically had a capacity of 1 to 6 terabytes, with the largest-capacity drives reaching 15 terabytes

3.2.2 RAM

Random access memory (RAM) is a type of data storage used in computers that is generally located on the motherboard. This type of memory is volatile and all information that was stored in RAM is lost when the computer is turned off.

The RAM chip may be individually mounted on the motherboard or in sets of several chips on a small board connected to the motherboard. Older memory types were in the form of chips called dual in-line package (DIP). Although DIP chips are still used today, the majority of memory is in the form of a module, a narrow printed circuit board attached to a connector on the motherboard. The three main memory circuit boards types containing chips are: RIMMs (Rambus in-line memory modules), DIMMs (dual in-line memory modules) and SIMMs (single in-line memory modules). Most motherboards today use DIMMs. Sometimes, people get confused about the difference between memory and storage, in part because both can be measured in megabytes (MB), gigabytes (GB) and terabytes (TB).

By contrast, storage is slower, but it can retain data when the device is powered down. So, for example, if a document has been saved to a hard drive prior to a power outage or system crash, the user will still be able to retrieve it when the system is back up and running.

3.2.3 Processor

A processor is an integrated electronic circuit that performs the calculations that run a computer. A processor performs arithmetical, logical, input/output (I/O) and other basic instructions that are passed from an operating system (OS). Most other processes are dependent on the operations of a processor.

Dual core is a CPU that has two distinct processors that work simultaneously in the same integrated circuit. This type of processor can function as efficiently as a single processor but can perform operations up to twice as quickly.

3.2.4 Mouse

A mouse is a small handheld input device that controls a computer screen's cursor or pointer in conjunction with the way it is moved on a flat surface. The mouse term name originates from its likeness to a small, corded and elliptical shaped device that looks like a mouse tail.

Some mouse devices have integrated features, such as extra buttons that may be programmed and assigned with different commands.

Because the mouse reduces the use of a keyboard, its invention and continuous innovation is considered one of the most important breakthroughs in computer ergonomics.

3.2.5 Keyboard

A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions. A keyboard is connected to a computer system using a cable or a wireless connection.

A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions.

3.2.6 Monitor

A computer monitor is an electronic device that shows pictures for computers. Monitors often look similar to televisions. The main difference between a monitor and a television is that a monitor does not have a television tuner to change channels. Monitors often have higher display resolution than televisions.

The advent of display technology has paved the way for the continuous evolution of

the monitor, whether it's for computers, television, mobile devices or any device that has a display. The current contenders for top-tier technology being used for display devices includes Super LCD 3 (SLCD3) and Super AMOLED. It should be noted that LED displays are actually just a kind of LCD display that use LED lights as backlight illumination.

CHAPTER 4

TESTING

4.1 System Testing

Quality assurance is an important step in software engineering. This overlaps with all the phases of development right from the requirement analysis. This quality requirement of the software system must be clearly extracted during the requirement analysis and all the subsequent phases should be made biased to that, the final testing will become trivial and less expensive. There are number of quality parameters like correctness, accuracy, reliability, robustness, efficiency, effectiveness, reusability, maintainability etc.. The state of requirement of each of these parameters will vary depending upon the name and domain of the application. The testing should be done at the end of all development steps. Even though the final testing and verification are inevitable for better life and functionality of the software. The different software testing approaches and methods like white box testing and black box testing. The major phases in testing are design of test plan, setting up test case and test candidate and test procedure, testing and correction. This is a cycle process and the software will circulate through all the steps till it attends the required quality. The testing is carried in the following steps.

4.2 Unit testing

Unit testing focuses verification effort on the smallest unit of software design the module. Using the details design description as a guide, important control paths are tested to uncover errors within boundary of the boundary of the module. The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing.

Unit testing is normally considered an adjacent to coding steps. After source level code has been developed, reviewed, and verified for correct syntax, unit test case design begins. A review of design information provides guidance for establishing test cases that are likely to uncover error in each case of the categories discussed above. Each test case should be coupled with a set of expected results.

4.3 Integration testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting test to uncover error associated with interfacing .The objective is to take unit tested modules and build a program structure that has been dictated by design.

There is often a tendency to attempt no incremental integration; that is to construct the program using a “big bang “approach. The entire modules are combined in advance. The entire program is tested as whole and chaoses usually result! A set of error is encountered. Correction is difficult because the isolation of cause is complicated by the vast expanse of entire program. Once errors are corrected, new ones appear and process continues in a seemingly endless loop.

CHAPTER 5

RESULT AND DISCUSSION, PERFORMANCE ANALYSIS

5.1 Result & Discussion

Journal publication helps to preserve your work in the permanent records of research in the field.

Adding your work to this record involves you in the active research community for a topic, helping to expand your professional network, increasing potential for collaboration and interaction with peer.

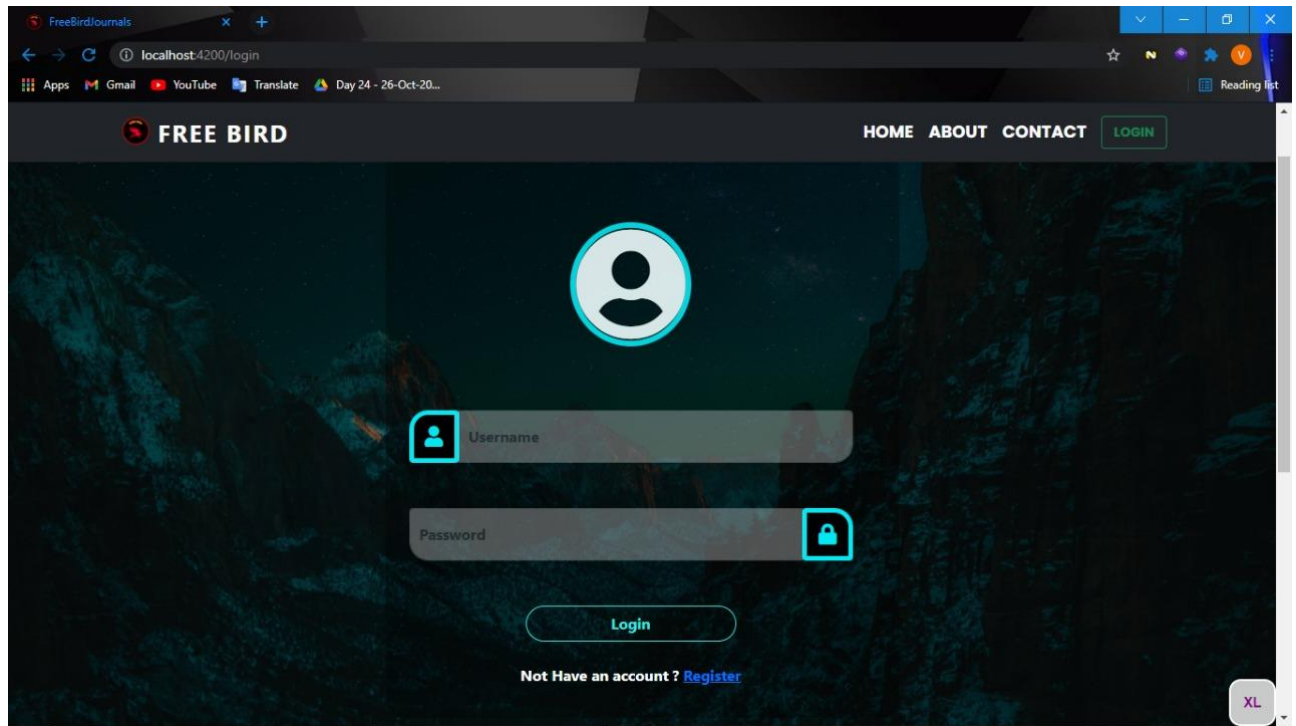
5.2 RECOMMENDATION

The Journal Publications meets a lot of expectations but would perform better if the following recommendations and suggestions are considered:

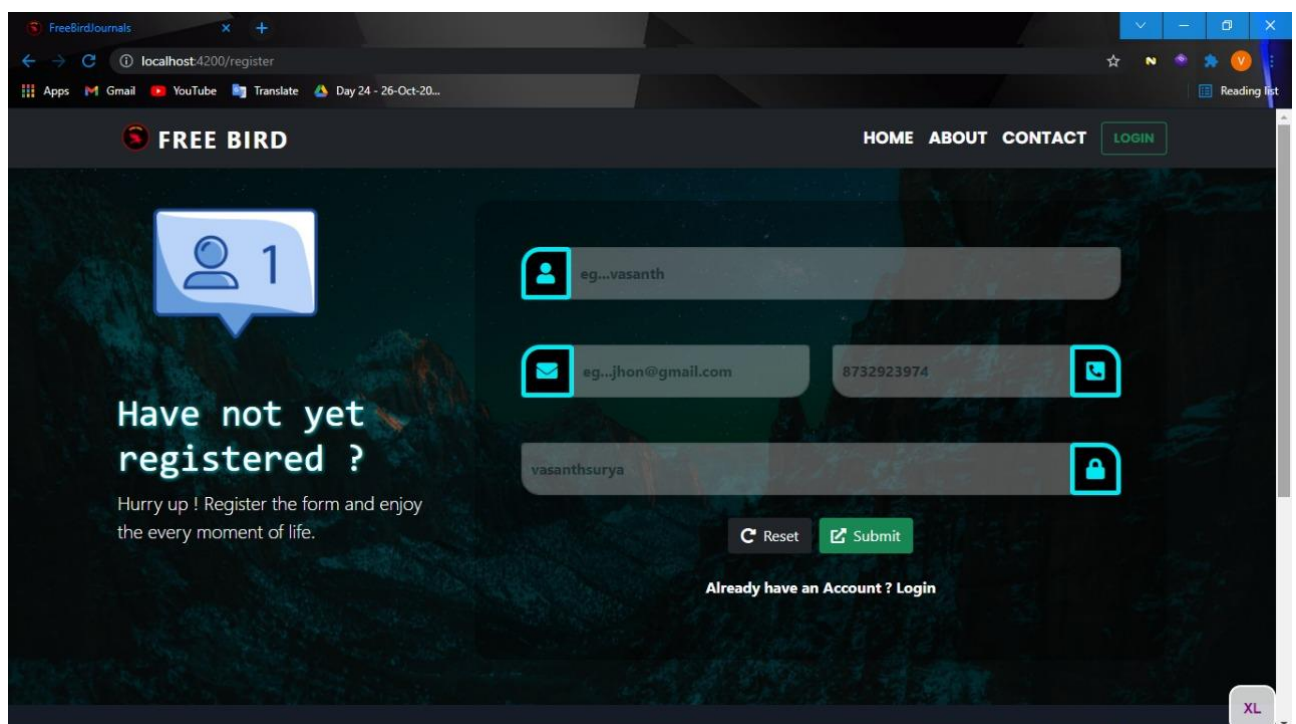
- i. System testing and maintenance should be performed regularly to avoid sudden system failures.
- ii. Updates and other modifications should be introduced with prior notice to users.
- iii. A course administrator should be employed that can manage the interactions between the donors, beneficiaries & administrators on one side and technicians on the other. This will aid ensure that proper support is provided for the system.

5.3 SCREENSHOTS

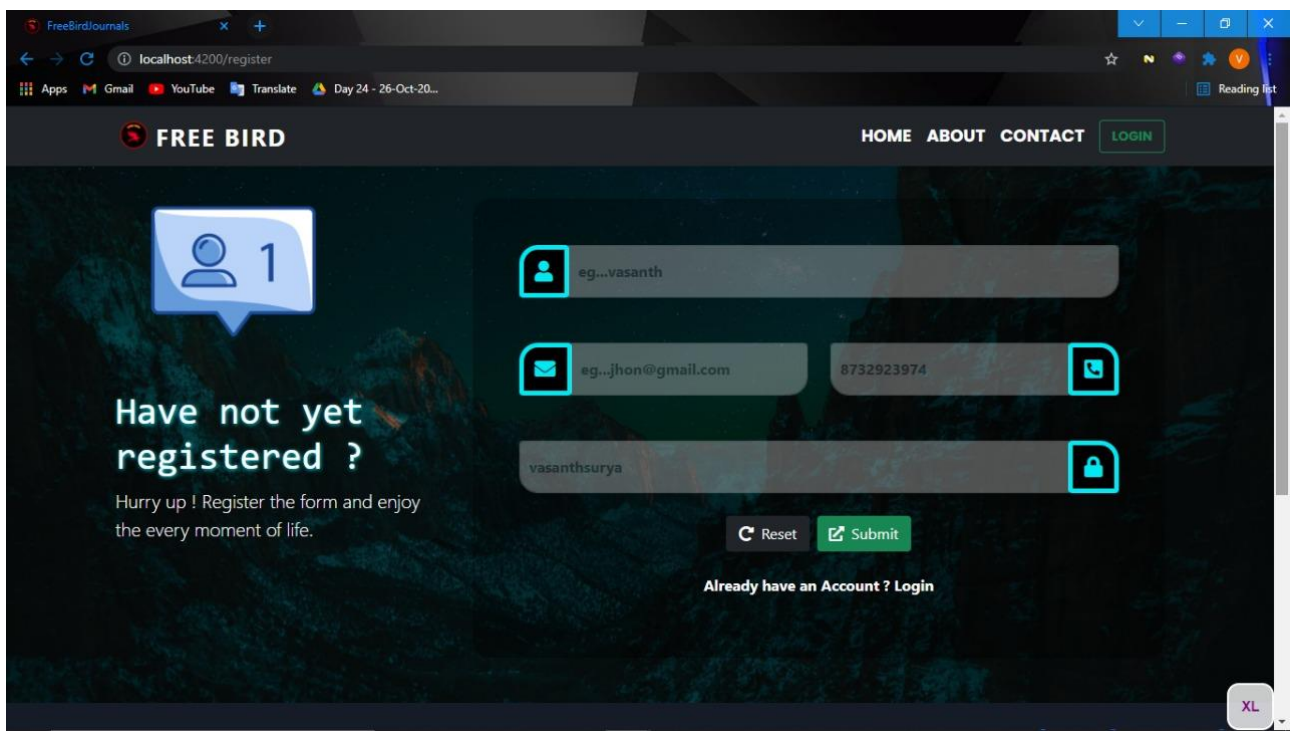
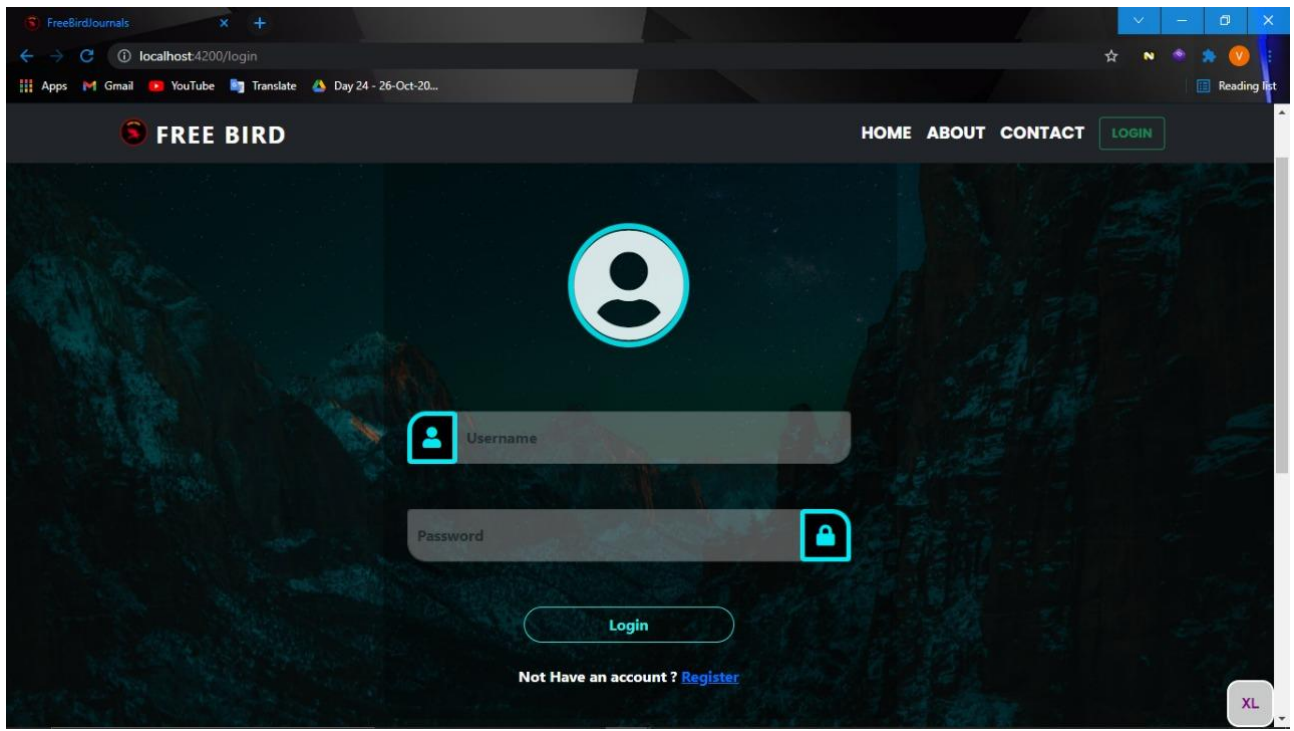
5.3.1 INPUT (Log-in page)

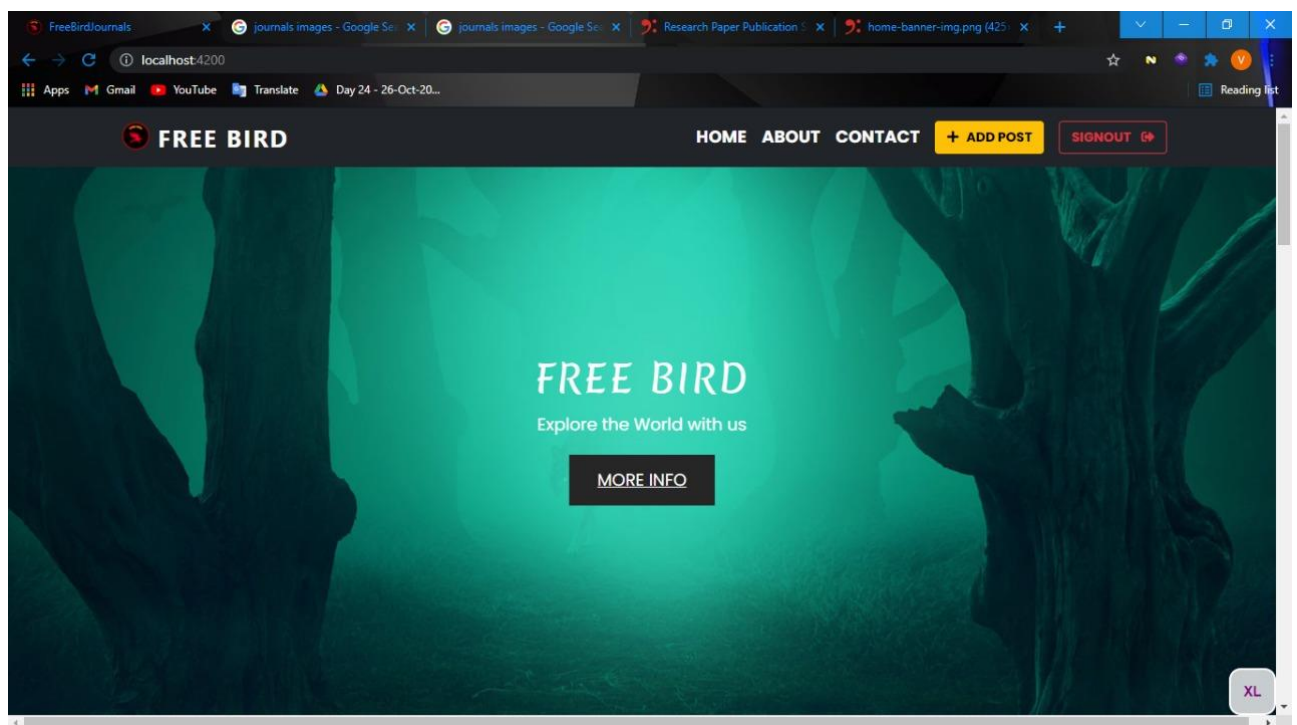
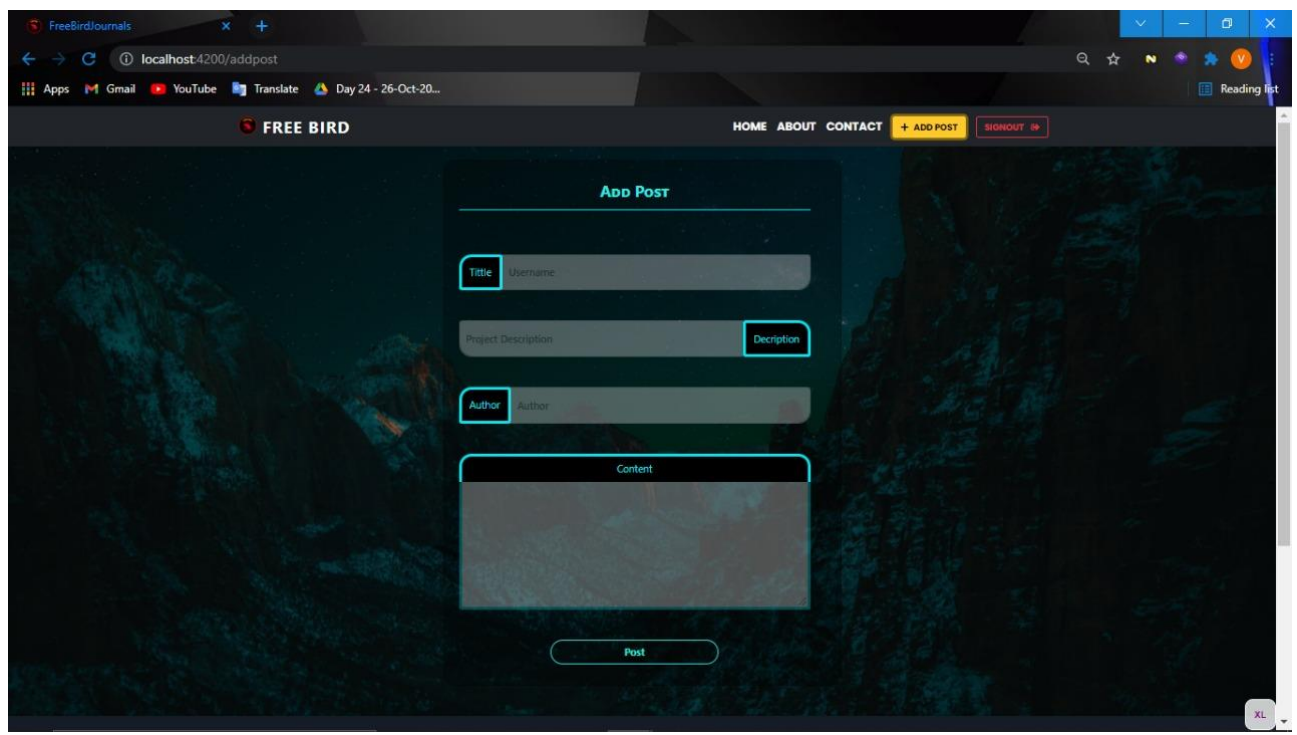


5.3.2 REGISTER PAGE



SAMPLE SCREENS






```
addpost.component.html X
src > app > addpost > addpost.component.html > div.hero > div.container > div.row > div.col-md-6.mx-auto.mt-4 > div.card.mb-5 > div.card-body.pb-5
32 |         <div class="input-group-prepend sym1">
33 |             <span class="input-group-text unmd"> Title </span>
34 |         </div>
35 |
36 |         <input type="text" class="form-control" name="ptitle" id="ptitle" aria-describedby="help1">
37 |     </div>
38 | </div>
39 |
40 | <!-- prjct desc -->
41 |
42 | <div class="form-group">
43 |     <div class="mb-5">
44 |         <div class="input-group ip2">
45 |             <input type="text" class="form-control" name="pdesc" id="pdesc" placeholder="Project Description" />
46 |             <div class="input-group-append sym2">
47 |                 <span class="input-group-text upd">Description</span>
48 |             </div>
49 |         </div>
50 |     </div>
51 | </div>
52 |
53 | <!-- post author -->
54 |
55 | <div class="form-group">
56 |     <div class="mb-5">
57 |         <div class="input-group ip1">
58 |             <input type="text" class="form-control" name="pauthor" id="pauthor" placeholder="Author" />
59 |             <div class="input-group-append sym3">
60 |                 <span class="input-group-text upd">Author</span>
61 |             </div>
62 |         </div>
63 |     </div>
64 | </div>
```

```
addpost.component.html X
src > app > addpost > addpost.component.html > div.hero > div.container > div.row > div.col-md-6.mx-auto.mt-4 > div.card.mb-5 > div.card-body.pb-5
64 |         <div class="input-group-prepend sym1">
65 |             <span class="input-group-text unmd"> Author </span>
66 |         </div>
67 |
68 |         <input type="text" class="form-control" name="pauthor" id="pauthor" aria-describedby="help1">
69 |     </div>
70 | </div>
71 |
72 | <!-- post content -->
73 |
74 | <div class="form-group">
75 |     <div class="mb-5">
76 |         <div class="mb-3">
77 |             <label for="" class="form-label unmd p-2 d-grid text-center">Content</label>
78 |             <textarea class="form-control txtar" name="mncnt" id="mncnt" rows="8" ngModel #pstcnt">
79 |         </div>
80 |     </div>
81 | </div>
82 |
83 | <!-- btn post -->
84 | <div class="mb-3 row text-center justify-content-center">
85 |     <div class="d-grid col-md-6">
86 |         <button type="button" class="btn lgbtn">Post</button>
87 |     </div>
88 | </div>
```

```

93         <div class="d-grid col-md-6">
94             <button type="button" class="btn btn-lg" >Post</button>
95         </div>
96     </div>
97
98
99     </form>
100
101
102
103
104
105
106
107
108     <!-- </form> -->
109
110 </div>
111 </div>
112 </div>
113 </div>
114
115 </div>

```

2. addpost.component.ts

```

TS addpost.component.ts X
src > app > addpost > TS addpost.component.ts > ...
1  import { Component, OnInit } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3  import { UserService } from '../user.service';
4
5  @Component({
6      selector: 'app-addpost',
7      templateUrl: './addpost.component.html',
8      styleUrls: ['./addpost.component.css']
9  })
10 export class AddpostComponent implements OnInit {
11
12     constructor(public usrSer : UserService) { }
13
14     msg: string;
15
16     ngOnInit(): void {
17     }
18
19     doAddPost(formdata : NgForm){
20
21
22
23         this.usrSer.addPost(formdata.value).subscribe((data : string) => {
24             console.log(data);
25             this.msg = data;
26
27         },(error : any) => {
28             console.log(error);
29             this.msg = "something went wrong";
30
31         });
32         formdata.reset( )
33

```

```

31     });
32     formdata.reset( )
33
34 }
35
36 }
37
38

```

3. editpost.component.html

```

editpost.component.html X
src > app > editpost > editpost.component.html > div.hero > div.container > div.row > div.col-md-6.mx-auto.mt-4 > div.card.mb-5 > div.card-body.pb-3
1 <div class="hero">
2   <div class="container">
3     <div class="row">
4       <div class="col-md-6 mx-auto mt-4">
5         <!-- card -->
6         <div class="card mb-5">
7           <!-- card header -->
8           <!-- card body -->
9           <div class="card-body pb-3 ps-4 pe-5">
10            <div class="cdhd text-center">
11              Edit Post
12            </div>
13            <form action="" #editpost="ngForm" (submit)=updpst(editpost) *ngIf="postData">
14              <p class="text-white">
15                {{msg}}
16              </p>
17              <!-- project title -->
18              <div class="form-group">
19
20                <div class="mb-5">
21                  <div class="input-group ip1">
22                    <div class="input-group-prepend sym1">
23                      <span class="input-group-text unm"> Title </span>
24                    </div>
25                    <input type="text" class="form-control" name="ptitle" id="ptitle" aria-describedby="hel
26                  </div>
27                </div>
28              </div>
29              <!-- prjct desc -->
30              <div class="form-group">
31                <div class="mb-5">
32                  <div class="input-group ip2">
33                    <input type="text" class="form-control" name="ndesc" id="ndesc" placeholder="Project De

```

```

editpost.component.html X
src > app > editpost > editpost.component.html > div.hero > div.container > div.row > div.col-md-6.mx-auto.mt-4 > div.card.mb-5 > div.card-body.pb-3
32 |         <div class="input-group ip2">
33 |             <input type="text" class="form-control" name="pdesc" id="pdesc" placeholder="Project De
34 |             <div class="input-group-append sym2">
35 |                 <span class="input-group-text upd">Description</span>
36 |             </div>
37 |         </div>
38 |     </div>
39 |
40 | </div>
41 | <!-- post author -->
42 | <div class="form-group">
43 |     <div class="mb-5">
44 |         <div class="input-group ip1">
45 |             <div class="input-group-prepend sym1">
46 |                 <span class="input-group-text unm"> Author </span>
47 |             </div>
48 |             <input type="text" class="form-control" name="pauthor" id="pauthor" aria-describedby="h
49 |         </div>
50 |     </div>
51 |
52 | </div>
53 | </div>
54 | <!-- post content -->
55 |
56 | <div class="form-group">
57 |     <div class="mb-5">
58 |         <div class="mb-3">
59 |             <label for="" class="form-label unmd p-2 d-grid text-center">Content</label>
60 |             <textarea class="form-control txtar" name="mncnt" id="mncnt" rows="8" [ngModel]="post
61 |         </div>
62 |     </div>
63 | </div>
64 | </div>
65 |
66 | <!-- btn post -->
67 | <div class="mb-3 row text-center justify-content-center">
68 |     <div class="d-grid col-md-6">
69 |         <button type="button" class="btn lgbt" >Post</button>
70 |     </div>
71 | </div>
72 | </form>
73 | <!-- </form -->
74 | </div>
75 | </div>
76 | </div>
77 |
78 | </div>

```

4. edit component.ts

```

TS editpost.component.ts X
src > app > editpost > TS editpost.component.ts > EditpostComponent > ngOnInit > activaterout.params.subscribe() callback
1 | import { Component, OnInit } from '@angular/core';
2 | import { NgForm } from '@angular/forms';
3 | import { ActivatedRoute, Params, Router } from '@angular/router';
4 | import { UserService } from '../user.service';
5 |
6 | @Component({
7 |     selector: 'app-editpost',
8 |     templateUrl: './editpost.component.html',
9 |     styleUrls: ['./editpost.component.css']
10 | })
11 | export class EditpostComponent implements OnInit {
12 |
13 |     constructor(public activaterout : ActivatedRoute , public usrSer : UserService) { }
14 |
15 |     postdata : {pid : string, ptitle : string, mncnt : string, pdesc : string , pauthor : string};
16 |
17 |     msg : string;
18 |
19 |     ngOnInit(): void {
20 |
21 |         this.activaterout.params.subscribe((param:Params) => {
22 |             console.log(param.postid);
23 |
24 |             if(param.postid){
25 |                 this.usrSer.getSinglePost(param.postid).subscribe((data:any[]) => {
26 |                     console.log(data);
27 |                     this.postdata = data[0];
28 |
29 |                 },(error) => {
30 |                     console.log(error);
31 |                 });
32 |             }
33 |         });

```

```

28
29
30     },(error) => {
31         console.log(error);
32     });
33 }
34
35
36 })
37 }
38
39 updpst(upd:NgForm){
40
41     upd.value.pid = this.postdata.pid
42     console.log(upd);
43
44     this.usrSer.editSinglePost(upd.value).subscribe((data:string) => {
45         this.msg = data;
46     },(error:any)=> {
47         this.msg = "somenting went wrong"
48     })
49 }
50 }
51
52
53 }

```

5. footerpg.component.html

```

<> footerpg.component.html X
src > app > footerpg > <> footerpg.component.html > ...
1      <!-- Footer -->
2      <footer
3          class="text-center text-lg-start text-white"
4          style="background-color: #1c2331"
5      >
6          <!-- Section: Social media -->
7          <section
8              class="d-flex justify-content-between p-4"
9              style="background-color: rgba(0, 0, 0, 0.2);"
10         >
11             <!-- Left -->
12             <div class="me-5">
13                 <span>Get connected with us on social networks:</span>
14             </div>
15             <!-- Left -->
16
17             <!-- Right -->
18             <div style="color: #03e9f4;">
19                 <a href="#" class="text-decoration-none me-4">
20                     <i class="fab fa-facebook-f"></i>
21                 </a>
22                 <a href="#" class="text-decoration-none me-4">
23                     <i class="fab fa-twitter"></i>
24                 </a>
25                 <a href="#" class="text-decoration-none me-4">
26                     <i class="fab fa-google"></i>
27                 </a>
28                 <a href="#" class="text-decoration-none me-4">
29                     <i class="fab fa-instagram"></i>
30                 </a>
31                 <a href="#" class="text-decoration-none me-4">
32                     <i class="fab fa-linkedin"></i>
33             </div>

```

```
❖ footerpg.component.html X
src > app > footerpg > ❖ footerpg.component.html > ...
32 </i class="fab fa-linkedin"></i>
33 </a>
34 <a href="" class=" text-decoration-none me-4">
35 <i class="fab fa-github"></i>
36 </a>
37 </div>
38 <!-- Right -->
39 </section>
40 <!-- Section: Social media -->
41
42 <!-- Section: Links -->
43 <section class="">
44 <div class="container text-center text-md-start mt-5">
45 <!-- Grid row -->
46 <div class="row mt-3">
47 <!-- Grid column -->
48 <div class="col-md-3 col-lg-4 col-xl-3 mx-auto mb-4">
49 <!-- Content -->
50 <h6 class="text-uppercase fw-bold">FreeBird Journals</h6>
51 <hr
52 class="mb-4 mt-0 d-inline-block mx-auto"
53 style="width: 60px; background-color: #03e9f4; height: 2px"
54 />
55 <p>
56 You can Explore many things with us and share your knowlegde with us....!
57 </p>
58 </div>
59 <!-- Grid column -->
60
61 <!-- Grid column -->
62
63 <!-- Grid column -->
64 <!-- nav -->
```

```
❖ footerpg.component.html X
src > app > footerpg > ❖ footerpg.component.html > ...
64 <!-- nav -->
65 <!-- Grid column -->
66 <div class="col-md-3 col-lg-2 col-xl-2 mx-auto mb-4">
67 <!-- Links -->
68 <h6 class=" fw-bold text-center" style="font-variant:small-caps">Free Bird</h6>
69 <!-- <hr
70 class="mb-4 mt-0 d-inline-block mx-auto"
71 style="width: 60px; background-color: #03e9f4; height: 2px"
72 /> -->
73 <p>
74 <!-- <a href="#" class="text-white">Your Account</a> -->
75 
76 </p>
77 <!-- <p>
78 <a href="#" class="text-white">Become an Affiliate</a>
79 </p>
80 <p>
81 <a href="#" class="text-white">Shipping Rates</a>
82 </p>
83 <p>
84 <a href="#" class="text-white">Help</a>
85 </p> -->
86 </div>
87 <!-- Grid column -->
88
89 <!-- Grid column -->
90 <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-md-0 mb-4">
91 <!-- Links -->
92 <h6 class="text-uppercase fw-bold">Contact</h6>
93 <hr
94 class="mb-4 mt-0 d-inline-block mx-auto"
95 style="width: 60px; background-color: #03e9f4; height: 2px"
96 />
```

```

src > app > footerpg > footerpg.component.html > ...
96         />
97         <p><i class="fas fa-home mr-3"></i> New York, NY 10012, US</p>
98         <p><i class="fas fa-envelope mr-3"></i> info@example.com</p>
99         <p><i class="fas fa-phone mr-3"></i> + 01 234 567 88</p>
100        <p><i class="fas fa-print mr-3"></i> + 01 234 567 89</p>
101        </div>
102        <!-- Grid column -->
103    </div>
104    <!-- Grid row -->
105</div>
106</section>
107<!-- Section: Links -->
108
109<!-- Copyright -->
110<div
111    class="text-center p-3"
112    style="background-color: rgba(0, 0, 0, 0.2)"
113>
114    © 2020 Copyright:
115    <a class="text-white text-decoration-none" href="#"
116        ><b>FreeBird</b></a>
117    >
118</div>
119<!-- Copyright -->
120</footer>
121<!-- Footer -->

```

6. headerpg.component.html

```

headerpg.component.html X
src > app > headerpg > headerpg.component.html > nav.navbar.navbar-expand-lg.navbar-light.bg-dark.fixed-top > div.container > div#navbarSupportedContent
1 <nav class="navbar navbar-expand-lg navbar-light bg-dark fixed-top">
2     <div class="container">
3         <a class="navbar-brand ps-1 ms-0 trk" href="#">
4             
5             Free Bird</a> <button aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"
6             <div class="collapse navbar-collapse" id="navbarSupportedContent">
7                 <ul class="navbar-nav ms-auto">
8                     <li class="nav-item active">
9                         <a class="nav-link" routerLink="/">Home</a>
10                    </li>
11                    <li class="nav-item">
12                        <a class="nav-link" href="#">About</a>
13                    </li>
14                    <!-- <li class="nav-item">
15                        <a class="nav-link" href="#">Portfolio</a>
16                    </li> -->
17                    <li class="nav-item">
18                        *ngIf="usrSer.isAdmin()"
19                        <a class="btn btn-outline-primary fw-light" routerLink="/postmng">Manage Post</a>
20                    </li>
21                    <li class="nav-item ms-2 lgt">
22                        *ngIf="!usrSer.isLoggedIn()"
23                        <a class="btn btn-outline-success" type="button" routerLink="/login">Login</a>
24                    </li>
25                    <li class="nav-item me-2 ms-2 lgt">
26                        *ngIf="usrSer.isLoggedIn()"
27                        <a class="btn btn-warning" type="button" routerLink="/addpost"><i class="fa fa-plus me-2" aria-hidden="
28                    </li>
29                    <li class="nav-item ms-2 lgt">
30                        *ngIf="usrSer.isLoggedIn()"
31                        <a type="button" class="btn btn-outline-danger" (click)=doLogout(); routerLink="/">Signout <i class="fa
32                    </li>
33                </ul>

```

```

30         *ngIf= usrSer.isLoggedIn() >
31         <a type="button" class="btn btn-outline-danger" (click)=doLogout(); routerLink="/">Signout <i class="fa
32         </li>
33     </ul>
34 </div>
35 </div>
36 </nav>
37

```

7. headerpg.component.ts

```

TS headerpg.component.ts X
src > app > headerpg > TS headerpg.component.ts > HeaderpgComponent > isadmin
1  import { Component, HostListener, OnInit } from '@angular/core';
2  import { Router } from '@angular/router';
3  import { UserService } from '../user.service';
4
5  // @HostListener('window:scroll', ['$event'])
6
7  // onWindowScroll() {
8  //     let element = document.querySelector('.navbar') as HTMLElement;
9  //     if (window.pageYOffset > element.clientHeight) {
10 //         element.classList.add('navbar-inverse');
11 //     } else {
12 //         element.classList.remove('navbar-inverse');
13 //     }
14 // }
15
16 // var scrollInterval : any ;
17 // var lastScroll = false;
18 // function myScroll() {
19 //     lastScroll = true;
20 //     if (!scrollInterval) {
21 //         //scroll has started, do some coloring
22 //         lastScroll = false;
23 //         scrollInterval = setTimeout(function() {
24 //             if (!lastScroll) {
25 //                 //scroll has ended, revert the coloring
26 //                 scrollInterval = clearInterval(scrollInterval);
27 //             }
28 //             lastScroll = false;
29 //         }, 100);
30 //     } else {
31 //         lastScroll = true;
32 //     }
33 // }

```

```

32 //    }
33 // }
34
35 @Component({
36   selector: 'app-headerpg',
37   templateUrl: './headerpg.component.html',
38   styleUrls: ['./headerpg.component.css']
39 })
40 export class HeaderpgComponent implements OnInit {
41
42   constructor(public usrSer : UserService, public myrout : Router) { }
43
44   isadmin = !!this.usrSer.isAdmin();
45
46
47   ngOnInit(): void {
48   }
49
50   doLogout(){
51     localStorage.clear();
52     this.myrout.navigateByUrl("/")
53   }
54
55 }
56 function onWindowScroll() {
57   throw new Error('Function not implemented.');
```

8. home.component.html

```

home.component.html X
src > app > home > home.component.html > div.container > div.row.mb-5 > div.col-md-11.mx-auto > div.row.cdnt > div.col-md-4 > div.card.cd1.ani
1 <!-- carousel contents starts -->
2 <div id="carouselExampleInterval" class="carousel slide carousel-fade" data-bs-ride="carousel">
3   <div class="carousel-inner">
4     <!-- first item -->
5     <div class="carousel-item active">
6       
7       <div class="carousel-caption d-md-block">
8         <h5 style="overflow-y: hidden;" class="animated bounceInRight d-none d-md-block" style="animation-delay: 1s">Fr
9         <h6 style="overflow-y: hidden;" class="animated bounceInRight d-sm-block d-md-none" style="animation-delay: 1s
10        <!-- lge dve -->
11        <p class="animated bounceInLeft d-none d-md-block" style="animation-delay: 2s">Explore the World with us</p>
12        <p class="animated bounceInRight d-none d-md-block" style="animation-delay: 3s"><a href="#">More Info</a></p>
13        <!-- sm dve -->
14        <p class="animated bounceInLeft d-md-none d-block desp" style="animation-delay: 2s">Explore the World with us<
15        <p class="animated bounceInRight d-md-none d-block" style="animation-delay: 3s"><a href="#" class="desb">More
16      </div>
17    </div>
18    <!-- second item -->
19    <div class="carousel-item">
20      
21      <div class="carousel-caption d-md-block">
22        <h5 style="overflow-y: hidden;" class="animated slideInDown d-none d-md-block" style="animation-delay: 1s">Free
23        <h6 style="overflow-y: hidden;" class="animated slideInDown d-md-none d-block" style="animation-delay: 1s">Free
24        <!-- large device -->
25        <p class="animated fadeInUp d-none d-md-block" style="animation-delay: 2s">Your gateway to world-class research
26        <p class="animated zoomIn d-none d-md-block" style="animation-delay: 3s"><a href="#">More Info</a></p>
27        <!-- sm dve -->
28        <p class="animated fadeInUp d-md-none d-block desp" style="animation-delay: 2s">Your gateway to world-class res
29        <p class="animated zoomIn d-md-none d-block" style="animation-delay: 3s"><a href="#" class="desb">More Info</a>
30      </div>
31    </div>
32    <!-- third item -->
33    <div class="carousel-item">

```

```

home.component.html X
src > app > home > home.component.html > div.container > div.row.mb-5 > div.col-md-11.mx-auto > div.row.cdnt > div.col-md-4 > div.card.cd1.ani
32 <!-- third item -->
33 <div class="carousel-item">
34   
35   <div class="carousel-caption d-md-block">
36     <h5 style="overflow-y: hidden;" class="animated zoomIn d-none d-md-block" style="animation-delay: 1s">Free Bird
37     <h6 style="overflow-y: hidden;" class="animated zoomIn d-md-none d-block" style="animation-delay: 1s">Free Bird
38     <!-- lg dve -->
39     <p class="animated fadeInLeft d-none d-md-block" style="animation-delay: 2s">Journals</p>
40     <p class="animated zoomIn d-none d-md-block" style="animation-delay: 3s"><a href="#">More Info</a></p>
41     <!-- sm dve -->
42     <p class="animated fadeInLeft d-md-none d-block desp" style="animation-delay: 2s">Journals</p>
43     <p class="animated zoomIn d-md-none d-block" style="animation-delay: 3s"><a href="#" class="desb">More Info</a>
44   </div>
45 </div>
46 </div>
47 <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleInterval" data-bs-slide="prev">
48   <span class="carousel-control-prev-icon d-none" aria-hidden="true"></span>
49   <span class="visually-hidden">Previous</span>
50 </button>
51 <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleInterval" data-bs-slide="next">
52   <span class="carousel-control-next-icon d-none" aria-hidden="true"></span>
53   <span class="visually-hidden">Next</span>
54 </button>
55 </div>
56 <!-- carousel content ends -->
57 <!-- main content starts -->
58 <!-- search bar starts -->
59 <div class="srchbox">
60   <div class="row">
61     <div class="mt-5 col-sm-5 mx-auto">
62       <div class="mb-5 mt-md-5 srchbx">
63         <form action="">
64         <div class="p-1 bg-light rounded rounded-pill shadow-sm mb-4">

```

```

home.component.html X
src > app > home > home.component.html > div.container > div.row.mb-5 > div.col-md-11.mx-auto > div.row.cdcnt > div.col-md-4 > div.card.cd1.an
65         <div class="input-group">
66             <input type="search" placeholder="What're you searching for?" aria-describedby="button-addon1"
67             <div class="input-group-append">
68                 <button id="button-addon1" type="submit" class="btn btn-link text-primary"><i class="fa fa-se
69             </div>
70         </div>
71     </div>
72 </form>
73 </div>
74 </div>
75 </div>
76 </div>
77 <!-- search bar end -->
78 <!-- card items starts articles-->
79 <div class="container">
80     <div class="d-flex justify-content-center fs-6" *ngIf="postssshown == false">
81         <p class="spinner-border text-primary "></p>
82     </div>
83     <!-- first row -->
84     <div class="row mb-5">
85         <div class="col-md-11 mx-auto">
86             <div class="row cdcnt">
87                 <!-- 1st card -->
88                 <div class="col-md-4 " *ngFor="let posts of posts">
89                     <div class="card cd1 animated bounceInRight" style="animation-delay: 1s; animation-duration: 2s
90                         <!-- ../assets/img/img/gpn.jpg -->
91                         
93
94                         <h5 class=" card-title mb-2">
95                             <!-- Free bird -->{{posts.ptitle}}
96                         </h5>

```

```

home.component.html X
src > app > home > home.component.html > div.container > div.row.mb-5 > div.col-md-11.mx-auto > div.row.cdcnt > div.col-md-4 > div.card.cd1.an
97         <div class="card-text mb-4">
98             <!-- Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed, dolores... -->
99             {{posts.mncnt | slice:0:128}}...
100         </div>
101         <a href="#" class="btn btnrdmr btn-outline-primary float-end d-flex">Read more</a>
102     </div>
103 </div>
104 </div>
105 </div>
106
107 <!-- 2nd card -->
108
109 <!-- <div class="col-md-4">
110     <div class="card animated bounceInRight" style="animation-delay: 1.1s; animation-duration: 2s;"
111
112         
113
114         <div class="card-body">
115             <h5 class=" card-title mb-2">
116                 Free bird
117             </h5>
118             <div class="card-text mb-4">
119                 Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed, dolores...
120             </div>
121             <a href="#" class="btn btnrdmr btn-outline-primary float-end d-flex">Read more</a>
122         </div>
123     </div>
124 </div> -->
125 <!-- 3rd card -->
126 <!-- <div class="col-md-4">
127     <div class="card animated bounceInRight" style="animation-delay: 1.2s; animation-duration: 2s;"
128
129         

```

```

home.component.html x
src > app > home > home.component.html > div.container > div.row.mb-5 > div.col-md-11.mx-auto > div.row.cdcnt > div.col-md-4 > div.card.cd1.an
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
    
    <div class="card-body">
      <h5 class=" card-title mb-2">
        Free bird
      </h5>
      <div class="card-text mb-4">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed, dolores....
      </div>
      <a href="#" class="btn btnrdmr btn-outline-primary float-end d-flex">Read more</a>
    </div>
  </div>
</div> -->
</div>
</div><br>
<!-- second row -->
<div class="row mb-5 mt-3">
  <div class="col-md-11 mx-auto">
    <div class="row cdcnt">
      <!-- 1st card -->
      <!-- <div class="col-md-4 ">
        <div class="card cd1 animated bounceInLeft" style="animation-delay: 1.2s; animation-duration:
        
        <div class="card-body">
          <h5 class=" card-title mb-2">
            Free bird

```

```

home.component.html x
src > app > home > home.component.html > div.container > div.row.mb-5 > div.col-md-11.mx-auto > div.row.cdcnt > div.col-md-4 > div.card.cd1.an
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
    Free bird
    </h5>
    <div class="card-text mb-4">
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed, dolores....
    </div>
    <a href="#" class="btn btnrdmr btn-outline-primary float-end d-flex">Read more</a>
  </div>
</div>
</div> -->
<!-- 2nd card -->
<!-- <div class="col-md-4">
  <div class="card animated bounceInLeft" style="animation-delay: 1.1s; animation-duration: 2s;">
    
    <div class="card-body">
      <h5 class=" card-title mb-2">
        Free bird
      </h5>
      <div class="card-text mb-4">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed, dolores....
      </div>
      <a href="#" class="btn btnrdmr btn-outline-primary float-end d-flex">Read more</a>
    </div>
  </div>
</div> -->
<!-- 3rd card -->

```

```

190         </div> -->
191
192         <!-- 3rd card -->
193
194         <!-- <div class="col-md-4">
195             <div class="card animated bounceInLeft" style="animation-delay: 1s; animation-duration: 2s;">
196
197                 
198
199                 <div class="card-body">
200
201                     <h5 class="card-title mb-2">
202                         Free bird
203                     </h5>
204                     <div class="card-text mb-4">
205                         Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed, dolores....
206                     </div>
207                     <a href="#" class="btn btnrdmr btn-outline-primary float-end d-flex">Read more</a>
208                 </div>
209             </div>
210         </div> -->
211     </div>
212 </div>
213 </div>
214 </div>
215 <!-- card items end articles-->
216 <!-- main content ends -->

```

9. home.component.ts

```

TS home.component.ts X
src > app > home > TS home.component.ts > HomeComponent > ngOnInit > subscribe() callback
1  import { Component, OnInit } from '@angular/core';
2  import { UserService } from '../user.service';
3
4  @Component({
5      selector: 'app-home',
6      templateUrl: './home.component.html',
7      styleUrls: ['./home.component.css']
8  })
9  export class HomeComponent implements OnInit {
10
11      constructor(public usrSer : UserService) { }
12
13      msg : string;
14
15      postssshown = false;
16
17      posts : any[] = [];
18
19      ngOnInit(): void {
20
21          this.usrSer.getPosts().subscribe((data:any[]) => {
22              console.log(data);
23
24              this.postssshown = true;
25              this.posts = data;
26          }, (error : any) =>{
27              this.msg = "something went wrong"
28          })
29      })
30  }
31
32

```

10. Loginpg.component.html

```
loginpg.component.html X
src > app > loginpg > loginpg.component.html > div.hero
1 <div class="hero">
2   <div class="container">
3     <div class="row">
4       <div class="col-md-6 mx-auto mt-4">
5
6         <!-- card -->
7
8         <div class="card ">
9
10
11
12         <!-- card body -->
13
14         <div class="card-body pb-3 ps-4 pe-5">
15
16           <!-- user icon -->
17
18           <div class="mb-3 mt-5 useric mx-auto d-flex justify-content-center align-content-center">
19             <!--  -->
20             <span class=" text-white icc "><i class="fa fa-user-circle" aria-hidden="true"></i></span>
21           </div>
22
23           <p class=" mb-5 text-danger fw-bold">
24             {{msg}}
25           </p>
26
27           <form #lgf="ngForm" (submit)="doLogin(lgf)" method="post">
28
29           <!-- username -->
30
31           <div class="form-group">
32
33           <div class="mb-5">
```

```
loginpg.component.html X
src > app > loginpg > loginpg.component.html > div.hero
31 <div class="form-group">
32
33   <div class="mb-5">
34     <div class="input-group ip1">
35       <div class="input-group-prepend sym1">
36         <span class="input-group-text unme"><i class="fa fa-user" aria-hidden="true"></i></span>
37       </div>
38
39       <input type="text" class="form-control" name="uname" id="uname" ngModel aria-describedby="uname" />
40     </div>
41
42   </div>
43 </div>
44
45 <!-- userpassword -->
46
47 <div class="form-group">
48
49   <div class="mb-5">
50     <div class="input-group ip2">
51       <input type="password" class="form-control" name="upswd" id="upswd" ngModel placeholder="Password" />
52       <div class="input-group-append sym2">
53         <span class="input-group-text upd"><i class="fa fa-lock" aria-hidden="true"></i></span>
54       </div>
55     </div>
56
57   </div>
58 </div>
59
60 <!-- button -->
61
62 <div class="mb-3">
```

```

61         <!-- button          -->
62
63         <div class="mb-3 row text-center justify-content-center">
64             <div class="d-grid col-md-6">
65                 <button type="submit" value="submit" class="btn lgbt" >Login</button>
66             </div>
67         </div>
68
69         <!-- registration page link          -->
70
71         <div class="mb-3 text-center ngpg fw-bold mt-4 ">
72             <span>Not Have an account ? <a routerLink="/register">Register</a> </span>
73         </div>
74     </form>
75 </div>
76 </div>
77 </div>
78 </div>
79
80 </div>
81 </div>

```

11. Login.component.ts

```

TS loginpg.component.ts X
src > app > loginpg > TS loginpg.component.ts > ...
1  import { Component, OnInit } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3  import { Router } from '@angular/router';
4  import { UserService } from '../user.service';
5
6  @Component({
7      selector: 'app-loginpg',
8      templateUrl: './loginpg.component.html',
9      styleUrls: ['./loginpg.component.css']
10 })
11 export class LoginpgComponent implements OnInit {
12
13     constructor(public userSer : UserService, public myrout : Router) { }
14
15     msg : string;
16
17     ngOnInit(): void {
18
19         if(this.userSer.isLoggedIn()){
20             this.myrout.navigateByUrl("/")
21         }
22     }
23
24     doLogin(form:NgForm){
25         console.log(form.value);
26
27         this.userSer.doUderLogin(form.value).subscribe((data : any[]) => {
28
29             console.log(data);
30
31             if(data.length==0)
32             {
33                 this.msg = "Invalid Credentials"

```

```

31         if(data.length==0)
32         {
33             this.msg = "Invalid Credentials"
34         }
35         else{
36             localStorage.setItem("loggeduser", data[0]._id)
37             this.myrouter.navigateByUrl("/")
38         }
39     }
40
41     //      this.msg = data;
42
43     // console.log(this.msg);
44
45     // this.myrouter.navigateByUrl("/login")
46
47
48
49     },(error : any) => {
50         console.log("something went wrong");
51
52
53
54     })
55
56 }
57
58

```

Postmng.component.html

```

<? postmng.component.html X
src > app > postmng > <? postmng.component.html > div.container.mt-5
1 <div class="container mt-5">
2   <div class="col-sm-4 mt-4 mb-5">
3     <!-- Search: <input type="text" #search (keyup)=doSearch(search.value)> &nbsp;
4
5     <a routerLink="/adduser">
6       <button class="btn btn-success">Add User</button>
7     </a> -->
8
9   </div><br>
10  <div class="col-md-12 mx-auto tbl">
11    <table class="table table-striped mt-4">
12      <thead class=" bg-dark text-warning text-center">
13        <tr>
14          <th scope="col">#</th>
15          <th scope="col">Title</th>
16          <th scope="col">Post Id</th>
17          <th scope="col">Author</th>
18          <th scope="col">Operations</th>
19        </tr>
20      </thead>
21      <tbody>
22
23        <tr *ngFor="let posts of posts; index as i;" class="text-center ">
24          <th scope="row">{{i+1}}</th>
25          <td>{{posts.ptitle}}</td>
26          <td>{{posts.pid}}</td>
27          <td>{{posts.pauthor}}</td>
28          <td>
29            <!-- /edituser/{{user._id}} -->
30            <a routerLink="/editpost/{{posts.pid}}"> <button class="btn btn-warning fw-bold">Edit <i class=" ms-2 fa
31              <button class="btn btn-outline-danger fw-bold"> Delete <i class="fa fa-trash ms-1" aria-hidden="true"><
32            </td>
33          </td>

```

```

32         </tr>
33     </tbody>
34 </table>
35 </div>
36 <!-- <p *ngIf="posts1.length==0">Not Records Found</p> -->
37 </div>
38 </div>
39
40
41

```

12. Postmng.component.ts

```

TS postmng.component.ts X
src > app > postmng > TS postmng.component.ts > ...
1  import { Component, OnInit } from '@angular/core';
2  import { UserService } from '../user.service';
3
4  @Component({
5      selector: 'app-postmng',
6      templateUrl: './postmng.component.html',
7      styleUrls: ['./postmng.component.css']
8  })
9  export class PostmngComponent implements OnInit {
10
11      constructor(public usrSer : UserService) { }
12
13      msg : string;
14      posts : any[] = [];
15      posts1 : any[] = [];
16
17      ngOnInit(): void {
18
19          this.usrSer.getPosts().subscribe((data:any[]) => {
20              console.log(data);
21
22              // this.postssshown = true;
23
24              this.posts = data;
25
26          }, (error : any) =>{
27
28              this.msg = "something went wrong"
29
30          })
31
32      }
33      doSearch(search:string){

```

```

9
10    })
11
12    }
13    doSearch(search:string){
14    this.usrSer.searchUsers(search).subscribe((data:any[])=>{
15        console.log(data);
16        this.posts1=data;
17    },(error:any)=>{
18        console.log(error);
19    })
20    }
21
22
23 }
24
25
26

```

13. Register.component.html

```

register.component.html x
src > app > register > register.component.html > div.hero
1  <div class="hero">
2      <div class="container">
3          <div class="row">
4              <div class="col-md-4 text-white msgim">
5                  Have not yet registered ?</h1>
7                  <p class=" lead ">Hurry up ! Register the form and enjoy the every moment of life.</p>
8              </div>
9              <div class="col-md-8 my-5">
10                 <div class="card">
11                     <div class="card-body p-5" id="cdb">
12
13                         <span>
14                             <!-- {{msg}} -->
15                         </span>
16                     <!-- #regform="ngForm" -->
17                     <!-- (submit)="doRegist(regform)" -->
18
19                     <form #regform="ngForm" (submit)="userReg(regform)" method="post">
20
21                         <!-- username -->
22
23                         <div class="row">
24
25                             <div class="col-md-12">
26                                 <div class="form-floating mb-5">
27
28                                     <!-- <div class="form-floating mb-5">
29
30                                     <input type="text" class="form-control" (blur)=verifyName(uname.value) ngModel
31                                     <label for="namereg" class=" fw-bold" >Name : </label>
32

```

```

register.component.html X
src > app > register > register.component.html > div.hero
31 <label for="name" eg class="fw-bold" />Name </label>
32
33 </div> -->
34
35 <div class="input-group ip1">
36   <div class="input-group-prepend sym1">
37     <span class="input-group-text unm"> <i class="fa fa-user" aria-hidden="tru
38   </div>
39
40   <!-- <input type="text" class="form-control" name="unmae" id="uname" aria-descr
41
42
43
44
45   <input type="text" class="form-control" (blur)=verifyName(uname.value) ngModel
46
47
48 </div>
49
50
51 <p *ngIf="msg22.length==21" class="text-success mt-2">
52   * {{msg22}}
53 </p>
54 <p *ngIf="msg22.length!=21" class="text-danger mt-2">
55   * {{msg22}}
56 </p>
57
58
59 <!-- {{uname.errors | json}} -->
60
61 <p class="alert text-danger alrt mt-3 mx-auto mb-0" *ngIf="uname.errors?.requir
62
63   * Enter UserName

```

```

register.component.html X
src > app > register > register.component.html > div.hero
63   * Enter UserName
64
65   </p>
66   <p class="alert text-danger alrt mt-3 mx-auto" *ngIf="uname.errors?.minlength 8
67
68   * Enter atleast 2 characters
69
70   </p>
71
72   <!-- {{uname.errors | json}} -->
73
74   <!-- <p class=" alert alert-danger" *ngIf="uname.errors?.required && uname.touched"
75
76   Enter UserName
77
78   </p>
79   <p class=" alert alert-danger" *ngIf="uname.errors?.minlength && uname.touched">
80
81   Enter atleast 2 characters
82
83   </p> -->
84
85 </div>
86 </div>
87
88 </div>
89
90 <!-- email and phone -->
91
92 <div class="row mb-5">
93
94

```

```

src > app > register > register.component.html > div.hero
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125

```

```

<!-- email -->

<div class="col-md-6">
  <div class="input-group ip1">
    <div class="input-group-prepend sym1">
      <span class="input-group-text un1"> <i class="fa fa-envelope" aria-hidden="true"></i>
    </div>
    <input type="email" class="form-control" name="emailreg" id="emailreg" aria-describedby="emailreg">
  </div>

  <!-- {{uemail.errors | json}} -->

  <p class="alert text-danger alert mt-3 mx-auto" *ngIf="uemail.errors?.required && uemail.errors?.required">
    * Enter Email
  </p>
  <p class="alert text-danger alert mt-3 mx-auto" *ngIf="uemail.errors?.email && uemail.errors?.email">
    * Enter a Valid Email
  </p>
</div>

<!-- phone -->

```

```

src > app > register > register.component.html > div.hero
126 <div class="col-md-6">
127   <div class="input-group ip2">
128     <input type="text" class="form-control" name="phonereg" id="phonereg" aria-describedby="phonereg" />
129     <div class="input-group-append sym2">
130       <span class="input-group-text upd"><i class="fa fa-phone-square-alt" aria-hidden="true"></i></span></div>
131   </div>
132   <div class="form-text">
133     <!-- {{uphone.errors | json}} -->
134   </div>
135   <div class="alert text-danger alert mt-3 mx-auto" *ngIf="uphone.errors?.pattern">
136     <div class="text-center">
137       <span>Please Enter a valid phone Number</span>
138     </div>
139   </div>
140 </div>
141
142 </div>
143
144 <!-- {{vname.name22}} -->
145
146 <!-- passwords -->
147
148 <div class="row">
149   <div class="col-md-12 mb-4">
150     <div class="input-group ip2">
151       <input type="password" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" class="form-control" />
152       <div class="input-group-append sym2">
153         <span class="input-group-text upd"><i class="fa fa-lock" aria-hidden="true"></i></span></div>
154     </div>
155   </div>
156 </div>
157

```

```

register.component.html X
src > app > register > register.component.html > div.hero
157     </div>
158     <!-- {{upswd.errors | json}} -->
159
160     <p class="alert text-danger alrt mt-3 mx-auto" *ngIf="upswd.errors?.pattern && upswd.errors?.pattern !== null" -->
161         * Please Enter atleast one uppercase and number . it should be more than 8 char
162     </p>
163
164 </div>
165
166 </div>
167
168 <!-- confirm password -->
169
170 <!-- <div class="col-md-6 ">
171
172     <div class="form-floating mb-3">
173
174         <div class="form-floating mb-3">
175
176             <input type="password" class="form-control" name="cupswd" id="cfpswdreg" aria-d
177             <label for="cfpswdreg" class="fw-bold" >Confirm Password : </label>
178
179         </div>
180
181     </div>
182
183 </div> -->
184
185 </div>
186
187 <!-- <div class=" mb-3 d-flex justify-content-center">
188
189     <div class="form-check">

```

```

register.component.html X
src > app > register > register.component.html > div.hero
188
189     <div class="form-check">
190         <input type="checkbox" class="form-check-input" name="trmsreg" id="trmsreg" value="checked" -->
191         <label class="form-check-label fw-bold lbl" for="trmsreg" >
192             I agree to Terms and Conditions
193         </label>
194     </div>
195
196 </div> -->
197
198 <div class="mb-2 text-center">
199
200
201     <button type="reset" class="btn btn-dark mb-3 me-2"> <i class="pe-2 fa fa-redo-alt "></i>R
202     <button value="submit" type="submit" class="btn btn-success mb-3" [disabled]="regform.invalid" -->
203
204 </div>
205
206 <div class="mb-2 text-center fw-bold">
207     <p>Already have an Account ? <a routerLink="/login">Login</a></p>
208
209 </div>
210
211 </form>
212
213 </div>
214
215 </div>
216
217 </div>
218
219 </div>
220 </div>

```

14. Register.component.ts

```
TS register.component.ts X
src > app > register > TS register.component.ts > ...
3 import { Router } from '@angular/router';
4 import { UserService } from '../user.service';
5
6 @Component({
7   selector: 'app-register',
8   templateUrl: './register.component.html',
9   styleUrls: ['./register.component.css']
10 })
11 export class RegisterComponent implements OnInit {
12
13
14   msg : string ;
15   msg22 : string ;
16
17   usernameavail : boolean;
18
19   constructor(public userser : UserService , public myrout : Router) { }
20
21   ngOnInit(): void {
22   }
23
24   userReg(form: NgForm){
25
26     console.log("user registered");
27     console.log(form.value);
28
29
30     this.userser.douserreg(form.value).subscribe((data : string) => {
31
32       console.log(data);
33
34       this.msg = data;
35     })
```

```
TS register.component.ts X
src > app > register > TS register.component.ts > ...
32 console.log(data);
33
34   this.msg = data;
35
36   console.log(this.msg);
37
38   // this.myrout.navigateByUrl("/login")
39
40
41
42   },(error : any) => {
43     console.log(error);
44   })
45   })
46   form.reset();
47   this.myrout.navigateByUrl("/login")
48 }
49
50 verifyName(unmae : string){
51
52   this.userser.userNameChecking(unmae).subscribe((data : string) => {
53
54     console.log(data);
55
56     if(data.length==0){
57       this.msg22 = "UserName is Available"
58       this.usernameavail = true;
59     }
60     else{
61       this.msg22 = "UserName is already Taken"
62       this.usernameavail = false;
63     }
64   })
65 }
```

```

src > app > register > TS register.component.ts > ...
63     }
64
65     // this.msg = data;
66
67     // console.log(this.msg);
68
69     // this.myrouter.navigateByUrl("/login")
70
71
72
73 },(error : any) => {
74     console.log(error);
75
76 })
77 // form.reset();
78 // this.myrouter.navigateByUrl("/login")
79 }
80
81 }
82
83
84

```

15. App-routing.module.ts

```

TS app-routing.module.ts X
src > app > TS app-routing.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { HeaderComponent } from './headerpg/headerpg.component';
4  import { HomeComponent } from './home/home.component';
5  import { LoginpgComponent } from './loginpg/loginpg.component';
6  import { RegisterComponent } from './register/register.component';
7  import { AddpostComponent } from './addpost/addpost.component';
8  import { AuthGuard } from './auth.guard';
9  import { PostmngComponent } from './postmng/postmng.component';
10 import { EditpostComponent } from './editpost/editpost.component';
11 import { PgnotfoundComponent } from './pgnotfound/pgnotfound.component';
12 import { RoleGuard } from './role.guard';
13
14 const routes: Routes = [
15     {path: "hd", component: HeaderComponent},
16     {path: "", component: HomeComponent},
17     {path: "register", component: RegisterComponent},
18     {path: "login", component: LoginpgComponent},
19     {path: "postmng", component: PostmngComponent, canActivate : [RoleGuard]},
20     {path: "editpost/:postid", component: EditpostComponent},
21     {path: "addpost", component: AddpostComponent, canActivate : [AuthGuard]},
22     {path: "**", component : PgnotfoundComponent}
23 ];
24
25 @NgModule({
26     imports: [RouterModule.forRoot(routes)],
27     exports: [RouterModule]
28 })
29 export class AppRoutingModule { }
30

```

16. App.component.html

```
TS app.component.spec.ts  app.component.html X
src > app > app.component.html > ...
1  <!-- header cmponent -->
2
3  <app-headerpg></app-headerpg>
4
5  <!-- router for all site -->
6
7  <router-outlet></router-outlet>
8
9  <app-footerpg></app-footerpg>
10
```

17. App.module.ts

```
TS app.component.spec.ts X  TS app.module.ts X
src > app > TS app.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3  import { HttpClientModule } from '@angular/common/http';
4
5  import { AppRoutingModule } from './app-routing.module';
6  import { AppComponent } from './app.component';
7  import { LoginpgComponent } from './loginpg/loginpg.component';
8  import { HeaderpgComponent } from './headerpg/headerpg.component';
9  import { HomeComponent } from './home/home.component';
10 import { FooterpgComponent } from './footerpg/footerpg.component';
11 import { RegisterComponent } from './register/register.component';
12 import { FormsModule } from '@angular/forms';
13 import { AddpostComponent } from './addpost/addpost.component';
14 import { PostmngComponent } from './postmng/postmng.component';
15 import { EditpostComponent } from './editpost/editpost.component';
16 import { PgnotfoundComponent } from './pgnotfound/pgnotfound.component';
17
18 @NgModule({
19   declarations: [
20     AppComponent,
21     LoginpgComponent,
22     HeaderpgComponent,
23     HomeComponent,
24     FooterpgComponent,
25     RegisterComponent,
26     AddpostComponent,
27     PostmngComponent,
28     EditpostComponent,
29     PgnotfoundComponent
30   ],
31   imports: [
32     BrowserModule,
33     HttpClientModule
```

```

30 ],
31 imports: [
32     BrowserModule,
33     HttpClientModule,
34     FormsModule,
35     // BrowserModule,
36     AppRoutingModule
37 ],
38 providers: [],
39 bootstrap: [AppComponent]
40 })
41 export class AppModule { }
42

```

18. Authguard.ts

```

TS app.component.spec.ts TS auth.guard.ts X
src > app > TS auth.guard.ts > ...
1  import { Injectable } from '@angular/core';
2  import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot, UrlTree } from '@angular/router';
3  import { Observable } from 'rxjs';
4  import { Router } from '@angular/router'
5  import { UserService } from './user.service'
6
7  @Injectable({
8      providedIn: 'root'
9  })
10 export class AuthGuard implements CanActivate {
11
12     constructor(public usrSer : UserService , public myrout : Router ){
13
14     }
15     canActivate():boolean {
16
17         if(!this.usrSer.isLoggedIn()){
18             this.myrout.navigateByUrl("/login")
19         }
20         return this.usrSer.isLoggedIn();
21     }
22
23 }
24

```

19. Index.html

```
TS app.component.spec.ts X index.html X
src > index.html > ...
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>FreeBirdJournals</title>
6   <base href="/">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="icon" type="image/x-icon" href="favicon.ico">
9   <link rel="stylesheet" href="/assets/css/bootstrap.min.css">
10  <link rel="stylesheet" href="/assets/js/bootstrap.bundle.min.js">
11  <link href="https://fonts.googleapis.com/css2?family=Merienda&display=swap" rel="stylesheet">
12  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
13  <!-- <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet" -->
14  <script src="https://kit.fontawesome.com/08aa97b044.js" crossorigin="anonymous"></script>
15  <link rel="stylesheet" href="/assets/css/bootstrap.min.css">
16  <link href="https://cdn.jsdelivr.net/npm/animate.css@3.7.2/animate.min.css" rel="stylesheet">
17 </head>
18 <body>
19   <app-root></app-root>
20 </body>
21 </html>
22
```

20. Server.js

```
TS app.component.spec.ts {} angular.json JS server.js X
JS server.js > ...
1 const http = require("http");
2
3 const app = require("./bknd/app")
4
5
6 const server = http.createServer(app)
7
8 server.listen(4444 , () => {
9   console.log("port no 4444");
10  // console.log);
11 });
```

21. Ts.config.json

```
TS app.component.spec.ts {} angular.json TS tsconfig.json X
tsconfig.json > {} compilerOptions
1  /* To learn more about this file see: https://angular.io/config/tsconfig. */
2  {
3    "compileOnSave": false,
4    "compilerOptions": {
5      "baseUrl": "./",
6      "outDir": "./dist/out-tsc",
7      "forceConsistentCasingInFileNames": true,
8      "strict": true,
9      "strictNullChecks": false,
10     "strictPropertyInitialization": false,
11     "noImplicitReturns": true,
12     "noFallthroughCasesInSwitch": true,
13     "sourceMap": true,
14     "declaration": false,
15     "downlevelIteration": true,
16     "experimentalDecorators": true,
17     "moduleResolution": "node",
18     "importHelpers": true,
19     "target": "es2017",
20     "module": "es2020",
21     "lib": [
22       "es2018",
23       "dom"
24     ],
25   },
26   "angularCompilerOptions": {
27     "enableI18nLegacyMessageIdFormat": false,
28     "strictInjectionParameters": true,
29     "strictInputAccessModifiers": true,
30     "strictTemplates": true
31   }
32 }
33
```

SUMMARY AND CONCLUSIONS

Journal Publishing is the **process of presenting a focused overview of a completed research study** that is published in a peer-reviewed, scholarly source. A journal article summary provides potential readers with a short descriptive commentary, giving them some insight into the article's focus.

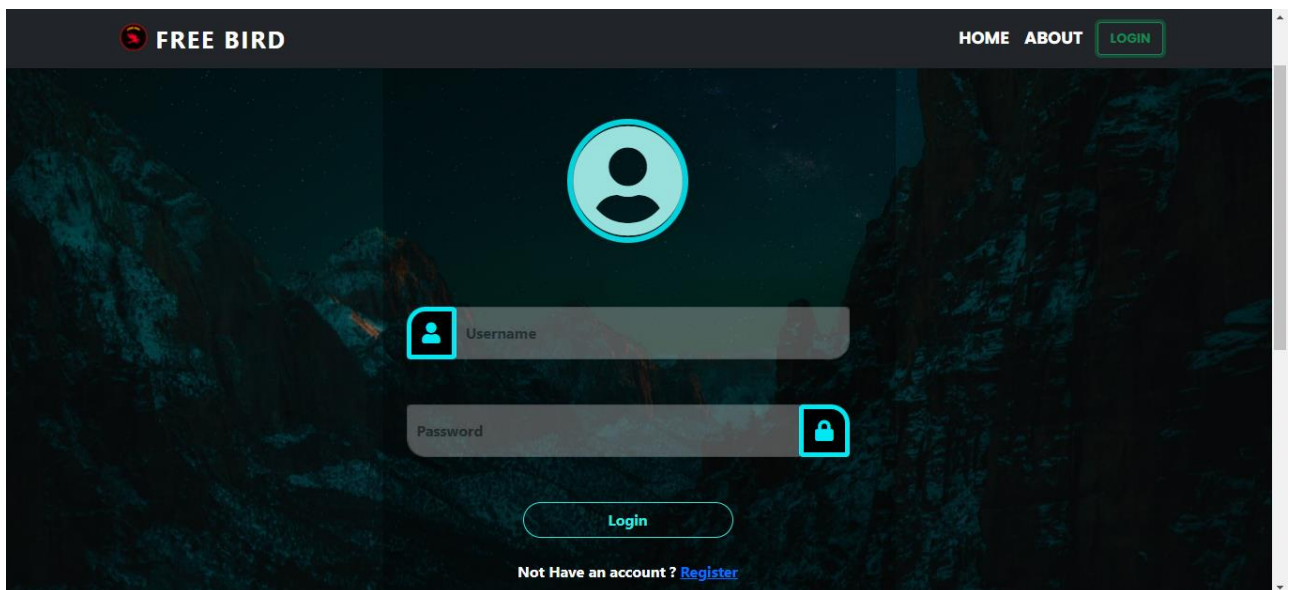
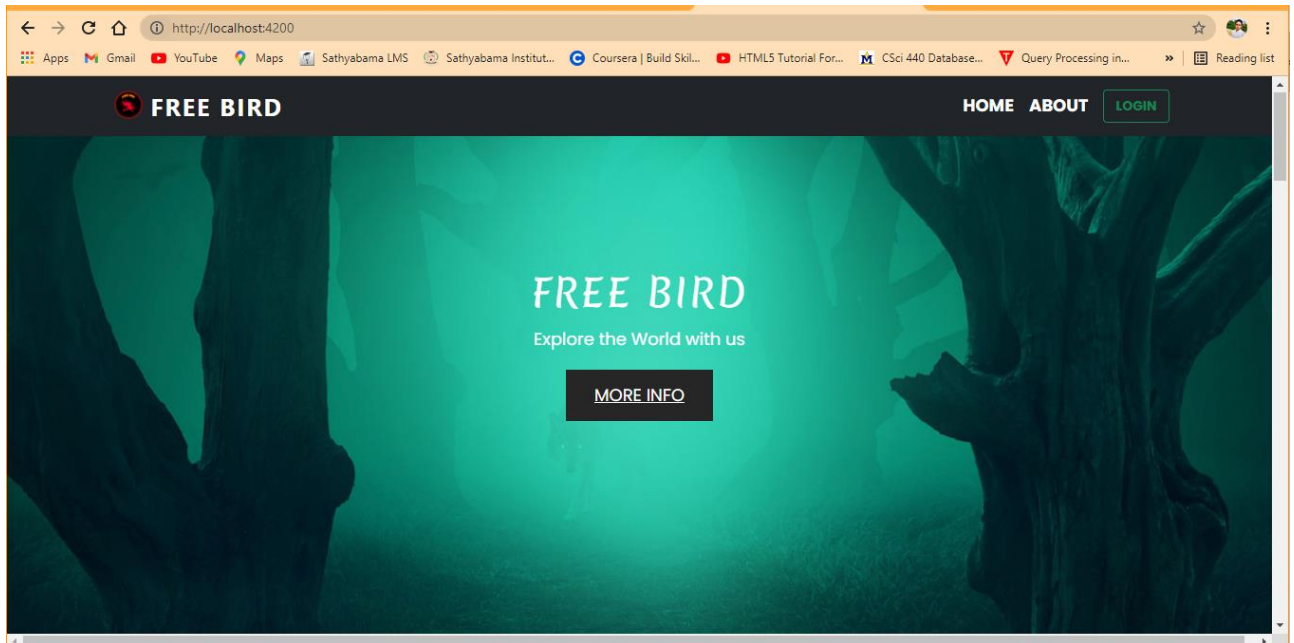
The system is highly user friendly and is well efficient to ease interactions with the users of the system. Reports generate with live data are proved to be informative and also helpful in making important decisions. The system is tested and implemented with high degree of accuracy.

The system is done with an insight into the necessary modification that may require in the future. Hence the system can be maintained successfully, without much rework.

REFERENCES

1. www.google.com
2. <https://developer.mozilla.org/en-US/>
3. www.pexels.com
5. www.wikipedia.org/angular
6. www.tutorialspointsg.com
7. www.javaworld.com
8. www.angularjs.com
9. www.nodejs.co

SAMPLE SCREENS



FREE BIRD

[HOME](#)
[ABOUT](#)
[LOGIN](#)

1

Have not yet registered ?

Hurry up ! Register the form and enjoy the every moment of life.

eg...vasanth

eg...jhon@gmail.com

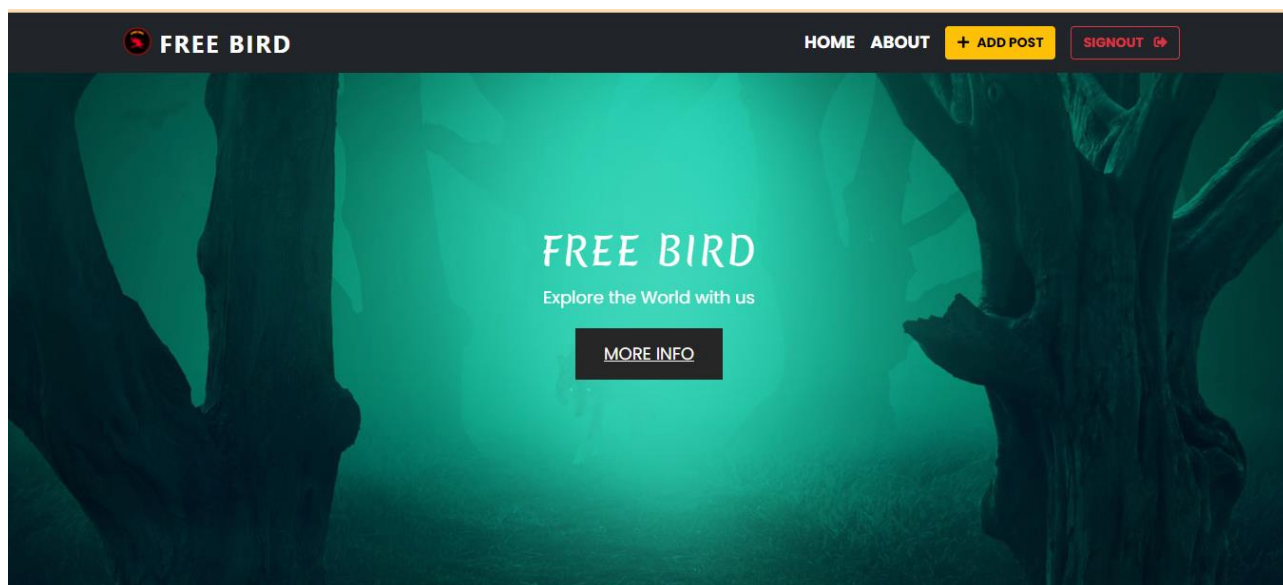
8732923974

vasanthasurya

Reset

Submit

Already have an Account ? [Login](#)



FREE BIRD

[HOME](#)
[ABOUT](#)
[+ ADD POST](#)
[SIGNOUT](#)

Reaction Chemistry & Engineering

Reaction engineering leverages the interface where fundamental molecular chemistry meets chemical engineering and technology. Ch...

[Read more](#)

Environmental Science: Advances

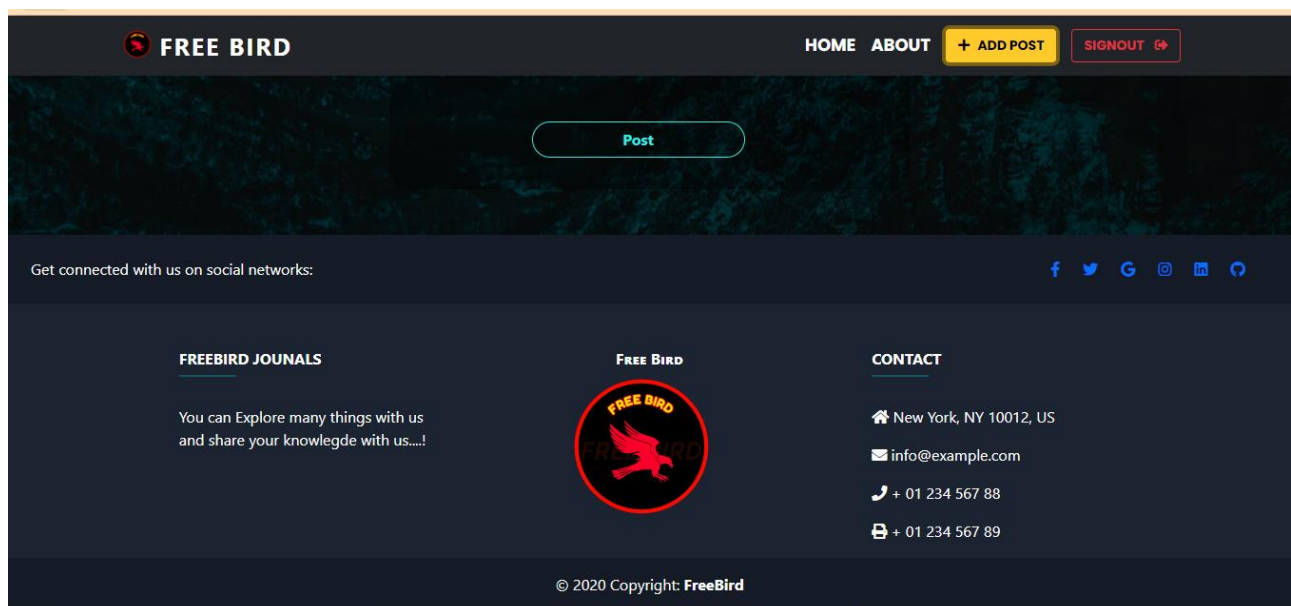
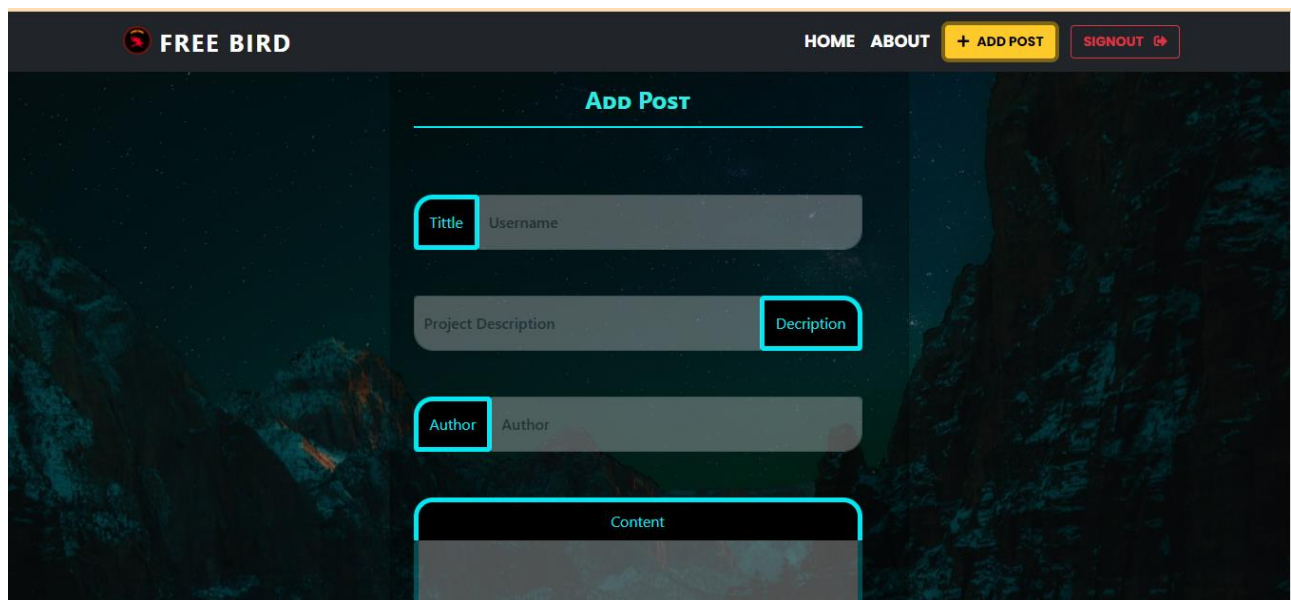
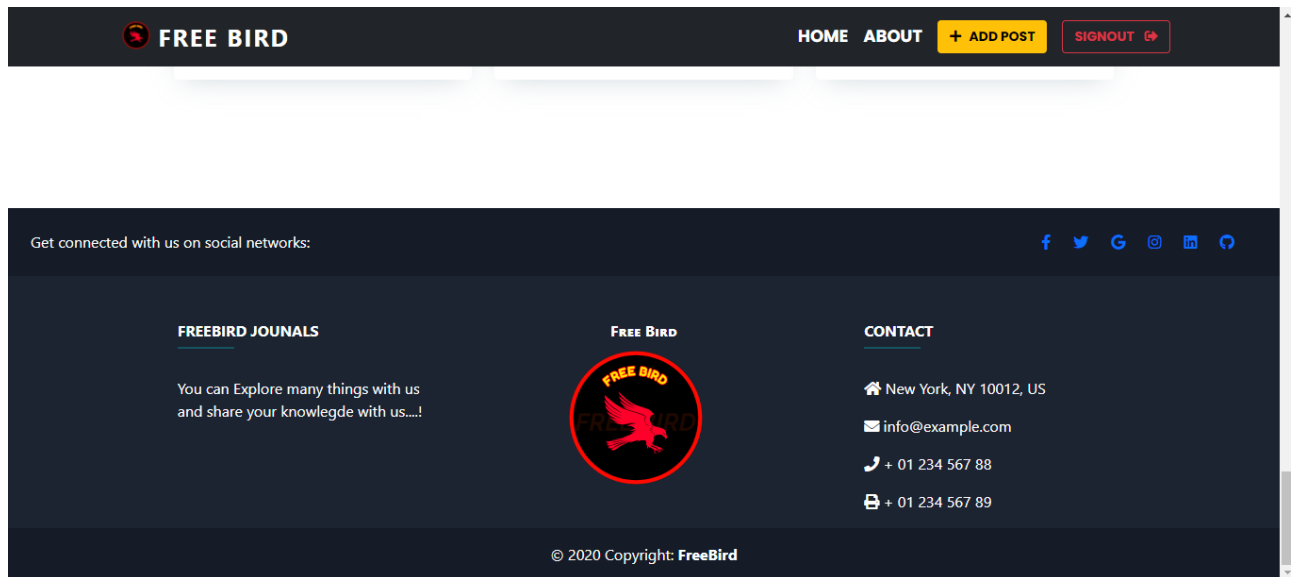
This journal is for anyone, in any discipline, working to advance environmental sustainability. Truly interdisciplinary, the jo...

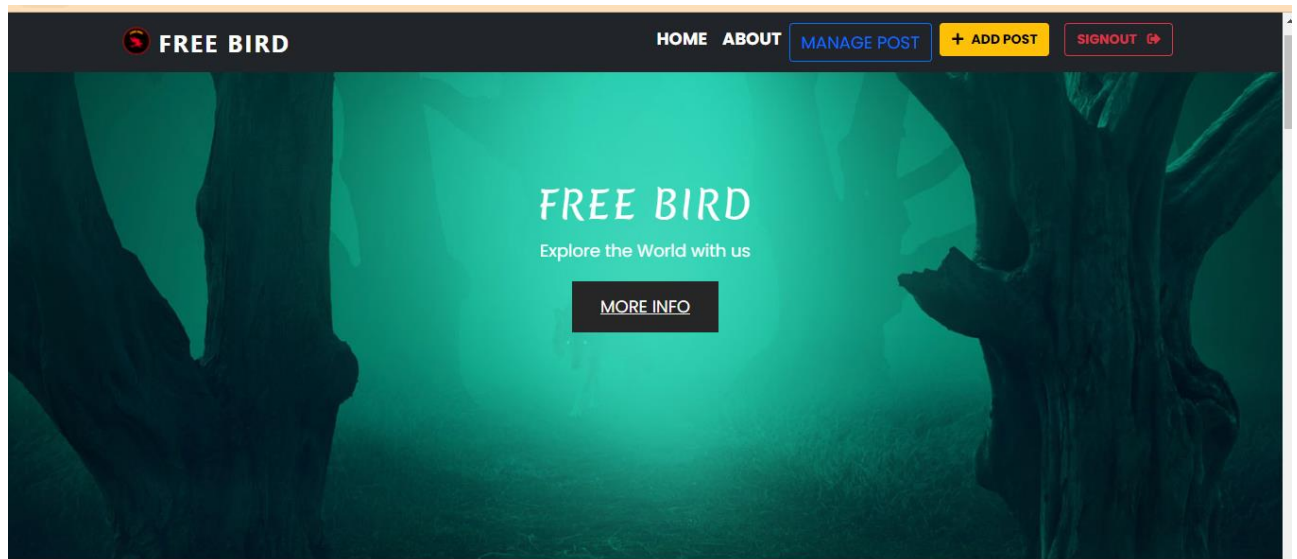
[Read more](#)

The Best Travel Journals to Bring on Your Next Adventure

Though we live in a digital age, there will always be something quite romantic and meditative about taking a moment to jot down ...

[Read more](#)





HOMEABOUTMANAGE POST+ ADD POSTSIGNOUT

Search:

#	Title	Post Id	Author	Operations	
1	Reaction Chemistry & Engineering	618eba27be353d19503b34f1	surya	Edit	Delete
2	Environmental Science: Advances	618ebbd2be353d19503b34f2	vasanth	Edit	Delete
3	The Best Travel Journals to Bring on Your Next Adventure	618ebdb6be353d19503b34f3	alan	Edit	Delete
4	DIGITAL HEALTH	618ebfcebe353d19503b34f4	jhon khan	Edit	Delete
5	The International Journal of Robotics Research	618ec0eabe353d19503b34f5	jason	Edit	Delete
6	Surgical Innovation	618ec199be353d19503b34f6	silver stallon	Edit	Delete