

PROFESSIONAL TRAINING REPORT - II

Entitled

Online Journal Publication Management System

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering with
specialization in Blockchain Technology

by

**GUMMADI VEERA MANOJ [Reg No : 41613010]
BATTU VARUN KUMAR [Reg No : 41613003]**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

**JEPPIAR NAGAR , RAJIV GANDHI SALAI ,
CHENNAI - 600119**

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Mr. GUMMADI VEERA MANOJ (41613010)** who carried out the project entitled "**ONLINE JOURNAL PUBLICATION MANAGEMENT SYSTEM**" under my supervision from January 2024 to May 2024.

Internal Guide

Ms. Lekshmi S Raveendran M.Tech.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 7/5/24

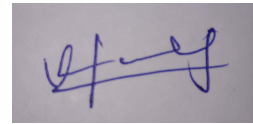
Internal Examiner

External Examiner

DECLARATION

I **GUMMADI VEERA MANOJ (41613010)**, hereby declare that the Professional Training Report-II entitled “**ONLINE JOURNAL PUBLICATION MANAGEMENT SYSTEM**” done by me under the guidance of **Ms Lekshmi S Raveendran M.Tech.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Blockchain Technology.

DATE: 07/05/2024



PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **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, **Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Ms Lekshmi S Raveendran M.Tech.**, for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 Professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

COURSE CERTIFICATE



Certificate of Completion

Is Awarded to

Gummadi Veera Manoj

Upon successfully completed the Bootcamp Training on
FULL STACK DEVELOPMENT for 40 hrs with a Mini Project in Online Art Gallery
from 11-Mar -2024 to 05-Apr -2024



A handwritten signature in black ink, likely belonging to Mr. Nikhil Barshikar.

Mr. Nikhil Barshikar
Managing Director
IMARTICUS LEARNING

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	vi
	LIST OF FIGURES	vii
1	INTRODUCTION	9
	1.1 Overview	
2	LITERATURE SURVEY	12
	2.1 survey	
3	REQUIREMENTS ANALYSIS	
	3.1 Objective	15
	3.2 3.2.1 Hardware Requirements	
	3.2.2 Software Requirements	16
4	DESIGN DESCRIPTION OF PROPOSED PRODUCT	22
	4.1.1 Ideation Map/Architecture Diagram	23
	4.1.2 Various stages	23
4.1	4.1.3 Internal or Component design structure	24
	4.1.4 working principles	28
	4.2 4.2.1 Novelty of the Project	33
5	RESULTS AND DISCUSSION	35
6	CONCLUSION	42
	REFERENCES	43
	APPENDIX	
	A. Research Paper	44
	B. Source Code	46

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1	System architecture	23
2	Project Flow	24
3	Login page	25
4	Login page working flow	27
5	Register page working flow	27
6	After login page (user page)	28
7	After login page (admin page)	28
8	Register page	30
9	Interface	30
10	Journal entries	31
11	Account list	31

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

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.

This project research titled “Design and Implementation of an Online Journal Management System which is a case study of Sokoto International Journal of Counselling Psychology (SIJCP)” is a web-based system which is aimed at using the potent ability of the Computer System to solve many of the challenges faced by the current manual system of Journal and Publication processes. It provides numerous benefits which include: Automation of the Submission process, review process, payment process and publication process. Contributors from any part of the world can submit their articles for assessment and publication, also reviewers can be assigned articles to review online without sending the hard copy to them and after the review process they can upload the reviewed article and fill the assessment form online for the contributor to view and download for corrections. Online payment was also integrated in the system using Vogue Pay Processor. With this system, Researchers, students, contributors can access all the published articles from any location in the world since the system is a web-based system. Payment options on the system and online review process need to be improved as indicated by the result of the analysis obtained after administering the usability questionnaire to some users of the system.

Scholarly publishing plays a critical role in the dissemination of knowledge and the advancement of academic research. It involves the publication of scholarly articles, research papers, and other academic works in journals, conference proceedings, and other publications. Scholarly publishing is a complex process that involves multiple stakeholders, including authors, editors, reviewers, publishers, and readers. Traditionally, the process of scholarly publishing has been

largely paper-based, with authors submitting manuscripts to journals by mail or in person, and editors and reviewers communicating by mail or telephone. However, with the advent of the internet and digital technologies, the process of scholarly publishing has undergone significant changes, leading to the development of online journal management systems (OJMS).

The development of OJMS can be traced back to the early 1990s when the internet became more widely accessible. Initially, OJMS were used primarily for online submission and peer review of manuscripts, allowing authors to submit their papers electronically and reviewers to access and review papers online. Over time, OJMS evolved to include a wide range of features and functionalities, such as automated manuscript tracking, online collaboration tools, and advanced analytics and reporting tools. Today, OJMS are used by thousands of scholarly journals around the world to manage the entire editorial process, from manuscript submission to publication.

An online journal management system (OJMS) is a web-based platform that facilitates the submission, review, and publication of scholarly articles and other academic works. OJMS are designed to streamline the editorial process, improve the efficiency and effectiveness of peer review, and enhance the overall quality of scholarly publications. OJMS typically include a range of features and functionalities, such as manuscript submission and tracking, peer review management, editorial workflow management, and content management. They also often include advanced analytics and reporting tools that provide editors and administrators with valuable insights into the journal's performance, submission trends, and reviewer behaviors.

The objective of this study is to provide a comprehensive overview of online journal management systems (OJMS), including their key features, advantages, challenges, and implications for scholarly publishing. The study aims to identify the key features of OJMS that distinguish them from traditional journal management systems and to explore the advantages and challenges of OJMS in the context of scholarly publishing. Additionally, the study seeks to discuss the implications of OJMS for the future of scholarly publishing and to propose recommendations for the future development and implementation of OJMS.

The study employs a mixed-methods approach, combining a literature review with a survey of journal editors and administrators who have experience with OJMS. The literature review examines existing research on OJMS, focusing on their key features, advantages, challenges, and implications for scholarly publishing. The survey collects data on the use of OJMS in scholarly publishing, including the types of OJMS used, their key features, and their perceived advantages and challenges. The data from the literature review and survey are analyzed to identify common themes and trends related to OJMS in scholarly publishing.

The study is organized into five main sections. The first section provides an overview of scholarly publishing and the evolution of online journal management systems. The second section defines and scopes OJMS, outlining their key features and functionalities. The third section discusses the objectives and methodology of the study. The fourth section presents the results of the literature review and survey, including the key features, advantages, challenges, and implications of OJMS. The fifth section discusses the results in the context of existing literature and proposes recommendations for the future development and implementation of OJMS. The study concludes with a summary of key findings and suggestions for further research. The proposed online journal management system (OJMS) represents a significant advancement in the field of scholarly publishing, offering a range of novel features and functionalities that distinguish it from existing systems. This essay will explore the key aspects of novelty in the proposed OJMS, including its integration of blockchain technology, the use of AI algorithms for manuscript screening and review assignment, and its advanced analytics and reporting tools.

The OJMS platform provided by us covers all aspects of online journal publishing, from establishing a journal website to operational tasks such as the author's submission process, peer review, editing, publication, archiving, and indexing of the journal. It also helps to manage the people aspects of organizing a journal, including keeping track of the articles, the work done by the editors, reviewers, and authors, notifying readers, and assisting with the correspondence.

CHAPTER 2

LITERATURE REVIEW

2.1 SURVEY

Open Science Framework (OSF): A comprehensive platform for managing research projects and sharing data. Zenodo: A general-purpose repository for research outputs, including project journals. Figshare: Another repository for various research outputs, with features for collaboration and version control. Several existing platforms offer online project journal publication functionalities. The Open Science Framework (OSF) provides a comprehensive suite of tools for managing research projects and sharing data, including project journals. Zenodo serves as a general-purpose repository for research outputs, where project journals can be deposited. Figshare offers another platform for research outputs, with features that support collaboration and version control for project journals.

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.

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 peerreviewed. 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.

Every journal these days needs an online application for the management of the journal works. In the previous decades, this was done by post and then it returned to email. But now an academic journal needs such a system to be able to remain in this arena. As of now, the authors would not send their papers to the journals by email or post. So, journal managers should think about this issue in the first days of launching new journals.

Many journals are managed by commercial publishers, or universities, these organizations always have already outsourced this job to an external technology company that is offering such a service. But, there are still some organizations and associations that do not offer such a service or their choice option is not suitable for the journal founders. Also, many journals are launching by private or independent scholars that does not have such a option and should decide about it on their own.

Aforementioned, these systems are of two types open source and commercials. The commercial systems are sold and supported by companies based on subscriptions and are supported and updated regularly. But their prices vary from very low to thousands of USD per year. It depends on the system and the add-ons that be purchased. As while it seems simple, a review of the current paradigms in journal management, reveals that theses systems are much more complicated than it seems.

The journal management systems are not now systems just to receive the manuscripts from the authors and pass them to the reviewers, they are much more than it. These systems now are constructed on a complicated infrastructure that links the publications of each journal to the archives, coding and identifying systems that are defined for the digital objects, the authors, and the organizations behind them. For example ScholarOne is one the famous web applications hosted by Clarivate institute, the owner of WoS Archive. Reviewing the services linked to the ScholarOne system reveals how much the extent of parallel systems for indexing, measuring the impact of publications, the indicators of the academic organizations, the

scientific indicators of the scholars, and connections between the archives are vast and multiple

llentrack is used for tracking manuscripts and accompanying correspondence through all stages of submission, preparation, and publication. The web-based application will also convert files from the major word processing formats to any of the major web-viewable formats.

The BenchPress is an online manuscript management system that includes the facilitation of peer review. It is customizable for each journal and will generate reports and statistics on the editorial process. BenchPress can prepare manuscripts for print or online publication.

Open Journal System (OJS) is a journal management and publishing software system developed by the Public Knowledge Project through its federally funded efforts to expand and improve access to research.

Available as open source, OJS assists with every stage of the refereed publishing process, from submissions to online publication and indexing. Through its management systems and its finely grained indexing of research, OJS seeks to improve the scholarly quality of refereed research.

This Online Journal Management System (OJMS) is used for tracking manuscripts and accompanying correspondence through all stages of submission, preparation, and publication. It is a web-based application will also convert files from the major word processing formats to any of the major web-friendly formats. It is a framework that offers proven solutions to many of the world's largest international publishers in areas such as information analysis, editorial and production.

Online Journal Management System (OJMS) is an easy-to-use online journal publishing and management platform that is designed for improved editorial management of scholarly journals. It also enables better accessibility to content. This OJMS drastically reduces the broadcasting duration of research papers by managing the publishing cycle

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

The objective of an online journal management system is to streamline and enhance the processes involved in managing academic journals. This includes submission, peer review, editing, publishing, and indexing of scholarly articles. The system aims to improve the efficiency, transparency, and quality of these processes, ultimately benefiting authors, reviewers, editors, and readers. Key objectives include:

1. ***Efficient Workflow:** Facilitate the smooth flow of submissions, reviews, revisions, and publications, reducing manual effort and administrative burden.
2. ***Peer Review Management:** Manage the peer review process effectively, ensuring timely and constructive feedback to authors while maintaining reviewer anonymity.
3. ***Author Services:** Provide authors with a user-friendly platform to submit manuscripts, track the status of their submissions, and communicate with editors.
4. ***Editorial Management:** Assist editors in managing the editorial process, assigning reviewers, making decisions on manuscripts, and overseeing the publication schedule.
5. ***Publication Ethics:** Ensure adherence to ethical standards in publishing, including plagiarism detection, authorship guidelines, and conflict of interest disclosures.
6. ***Indexing and Archiving:** Ensure proper indexing of journals in relevant databases and long-term archiving to preserve scholarly content.
7. ***Accessibility and Visibility:** Enhance the accessibility and visibility of journals by providing online access to a global audience, potentially increasing citations and impact.
8. ***Analytics and Reporting:** Provide tools for tracking journal performance, including submission rates, acceptance rates, publication times, and reader engagement metrics.

Overall, the objective is to leverage technology to improve the efficiency, transparency, and quality of academic publishing, benefiting all stakeholders in the scholarly community.

3.2 REQUIREMENTS

SOFTWARE REQUIREMENT:

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

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.

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. NonBlocking. 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 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.

Angular CLI

This topic 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 singledeveloper projects to enterprise-leve



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.

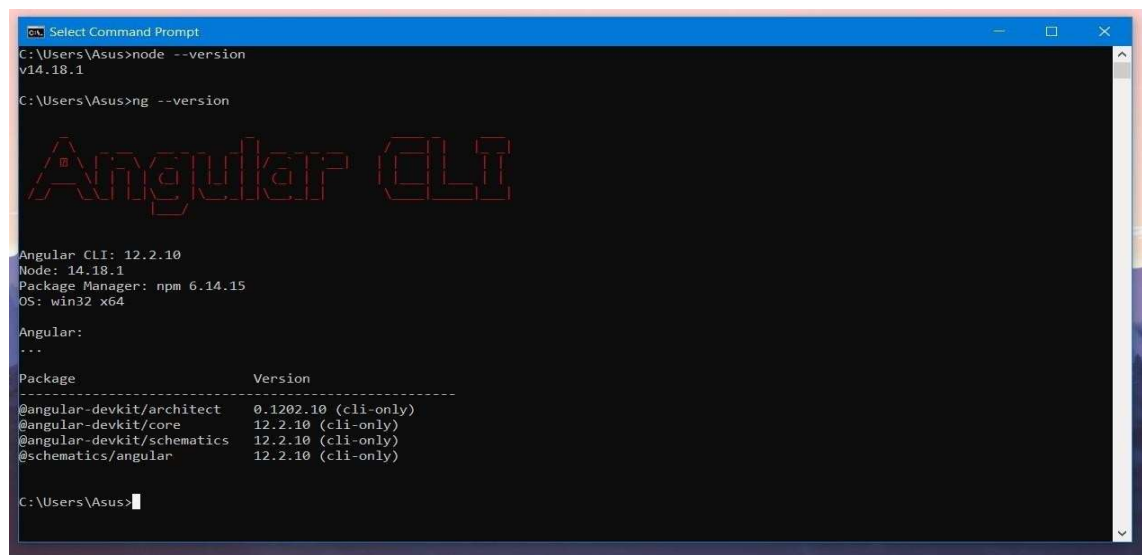
Components

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.

- Windows 10 or above



```

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>
```

- Node JS
- Angular CLI

- Database

3.2.1 HARDWARE REQUIREMENTS

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

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

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.

3.1.1 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.

3.1.2 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.

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.

3.1.3 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.

SOFTWARE REQUIREMENTS

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

CHAPTER 4

DESIGN DESCRIPTION OF PROPOSED PROJECT

4.1 PROPOSED METHODOLOGY

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.

4.1.1 Ideation Map/System Architecture

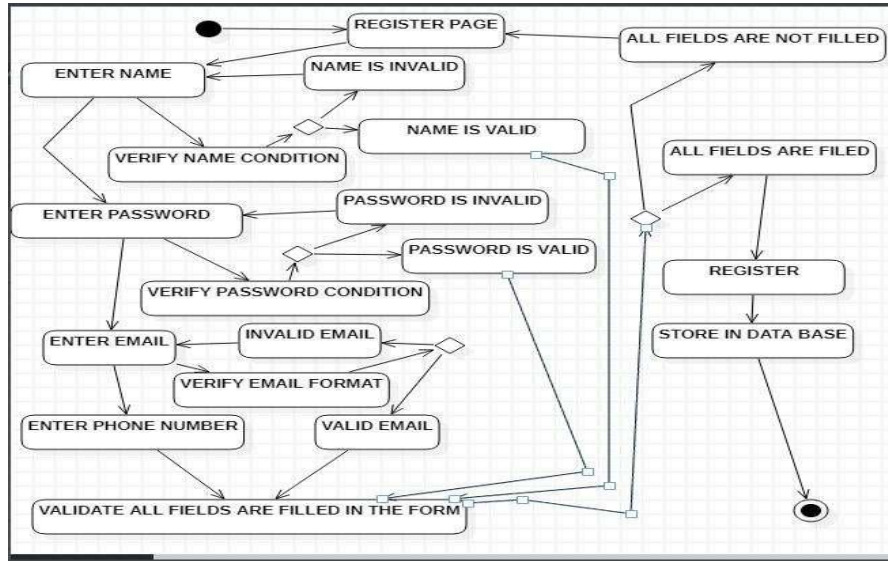


Fig (1) system architecture

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

4.1.2 Various Stages

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

4.1.3 Internal or Component design structure

PROJECT FLOW :

MAIN PAGE

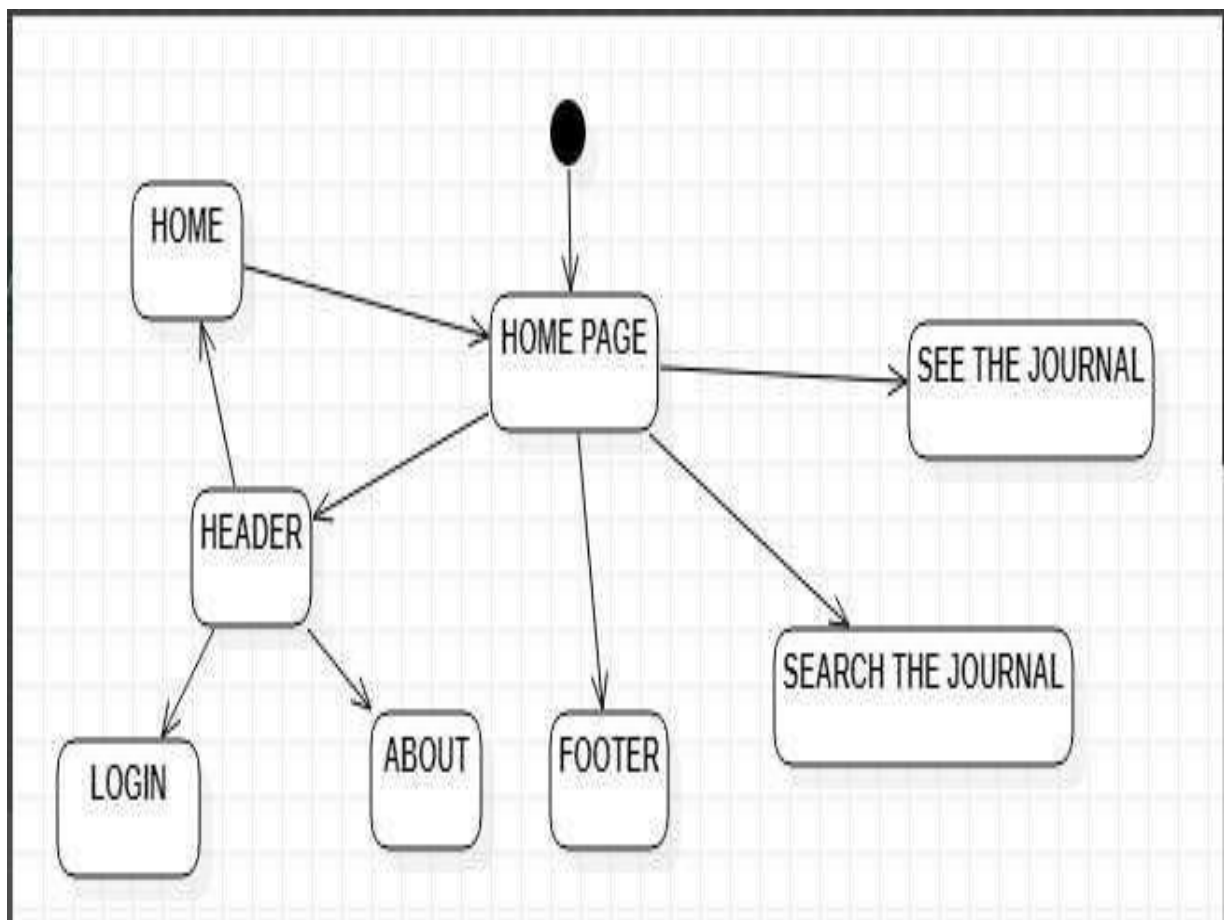


Fig (2) project flow

LOGIN PAGE

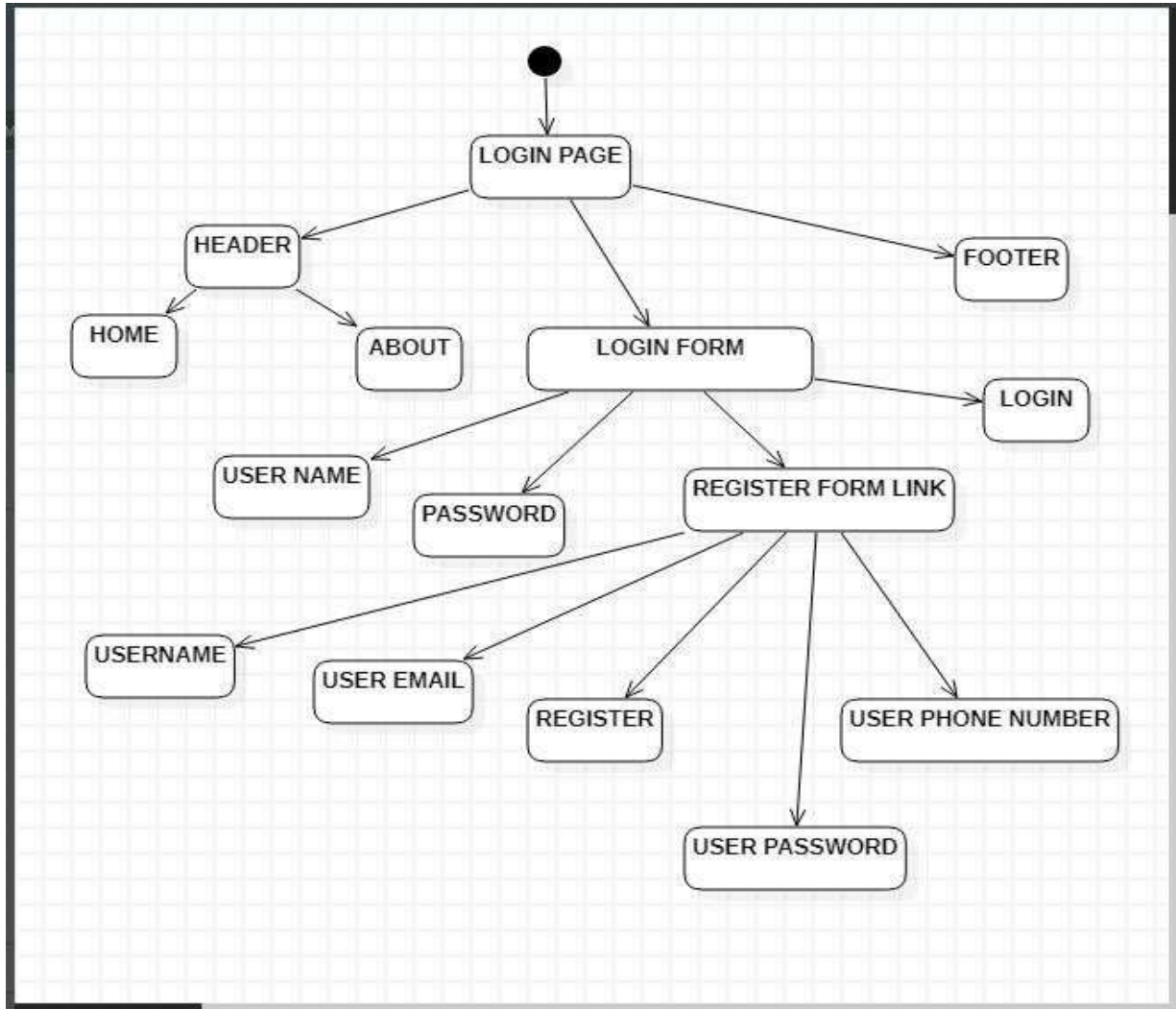
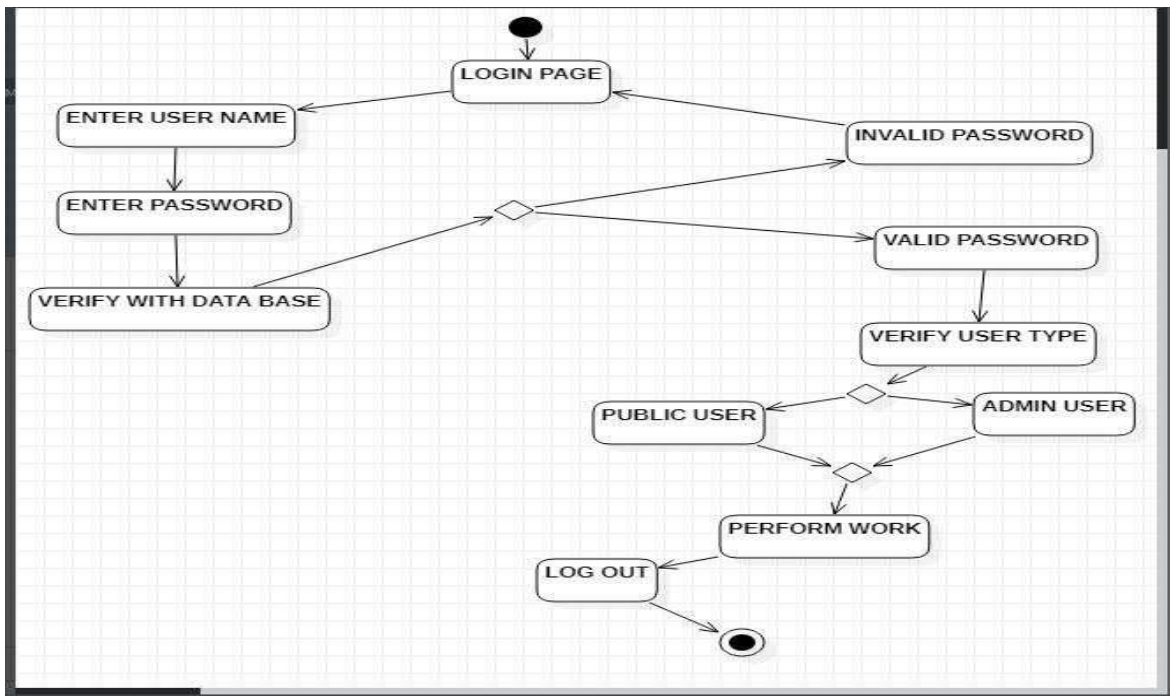


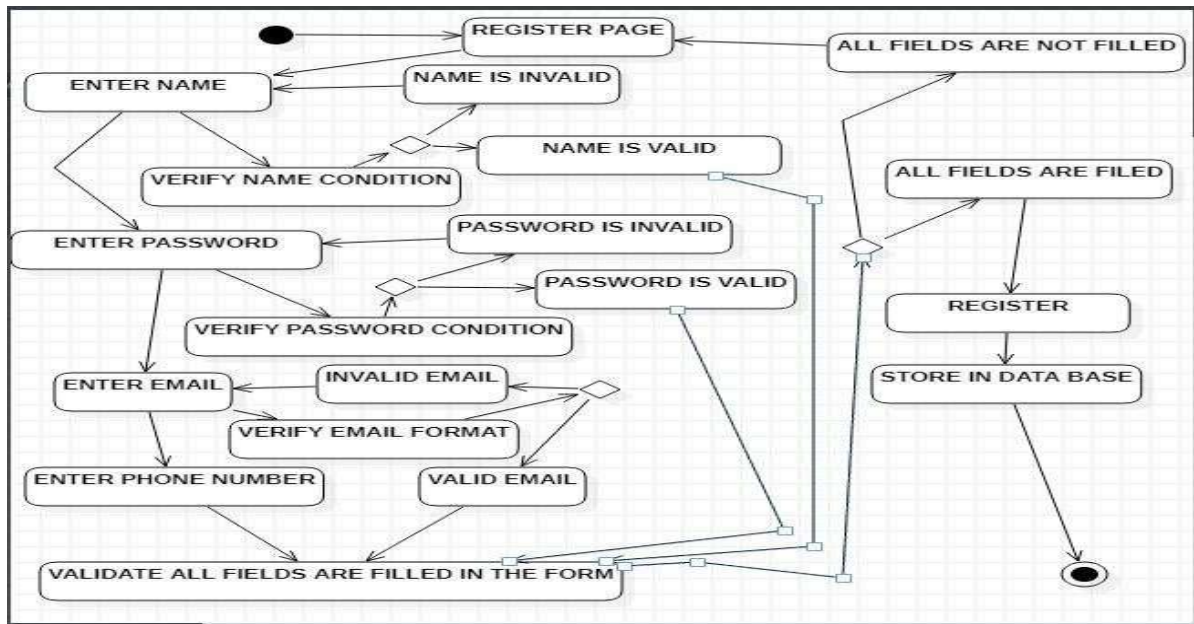
Fig (3) login page

LOGIN PAGE WORKING FLOW



Fig(4) Login page working flow

REGISTER PAGE WORKING FLOW



Fig(5) register page working flow

AFTER LOGIN PAGE (USER PAGE)

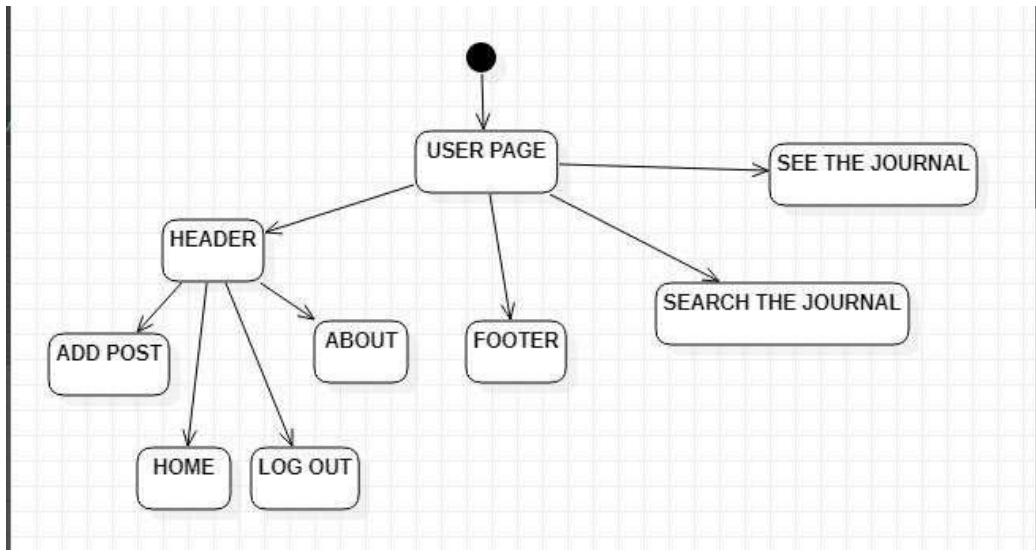


Fig (6) after login page (user page)

AFTER LOGIN PAGE (ADMIN PAGE)

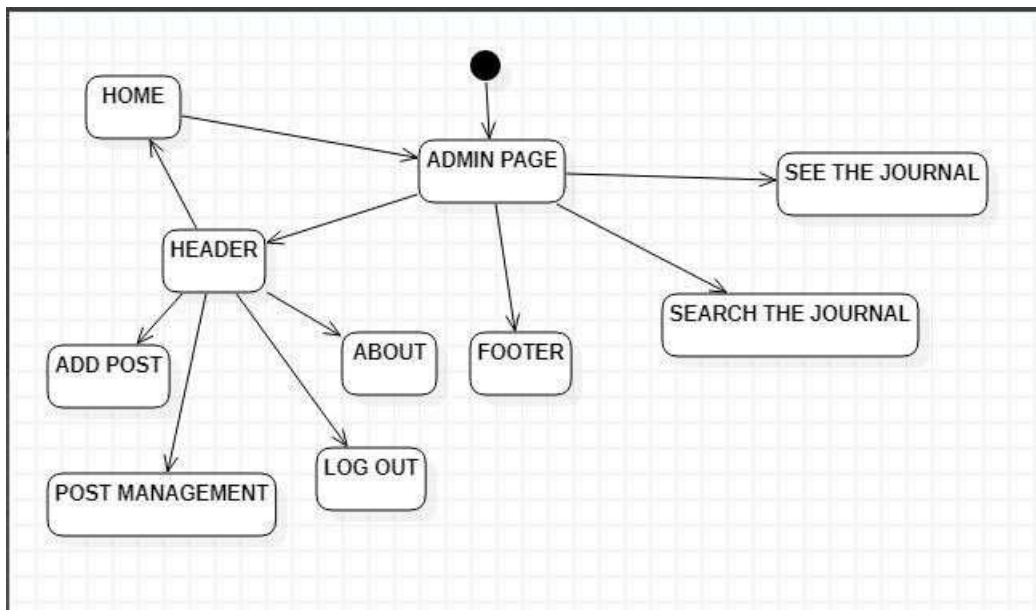


Fig (7) after login page (admin page)

4.1.4 working principles

Submission: Authors submit their manuscripts through the system, providing metadata and other required information.

Review Process: Editors assign submitted manuscripts to reviewers. Reviewers evaluate the manuscript and provide feedback to the editor through the system.

Decision Making: Based on the reviews, editors make a decision on whether to accept, reject, or request revisions to the manuscript. They communicate this decision to the authors through the system.

Revision and Resubmission: If revisions are requested, authors make changes to the manuscript and resubmit it through the system.

Publication: Once a manuscript is accepted, it goes through a final check and is scheduled for publication. The published article is then available online through the journal's website.

Archiving: The system archives published articles, making them accessible for future reference.

User Management: The system manages user roles and permissions, ensuring that only authorized users can perform certain actions, such as submitting manuscripts or making editorial decisions.

Overall, an online journal management system streamlines the process of managing scholarly articles, from submission to publication, while also providing a platform for communication between authors, reviewers, and editors.

4.1.5 Implementation Screenshots

REGISTER PAGE

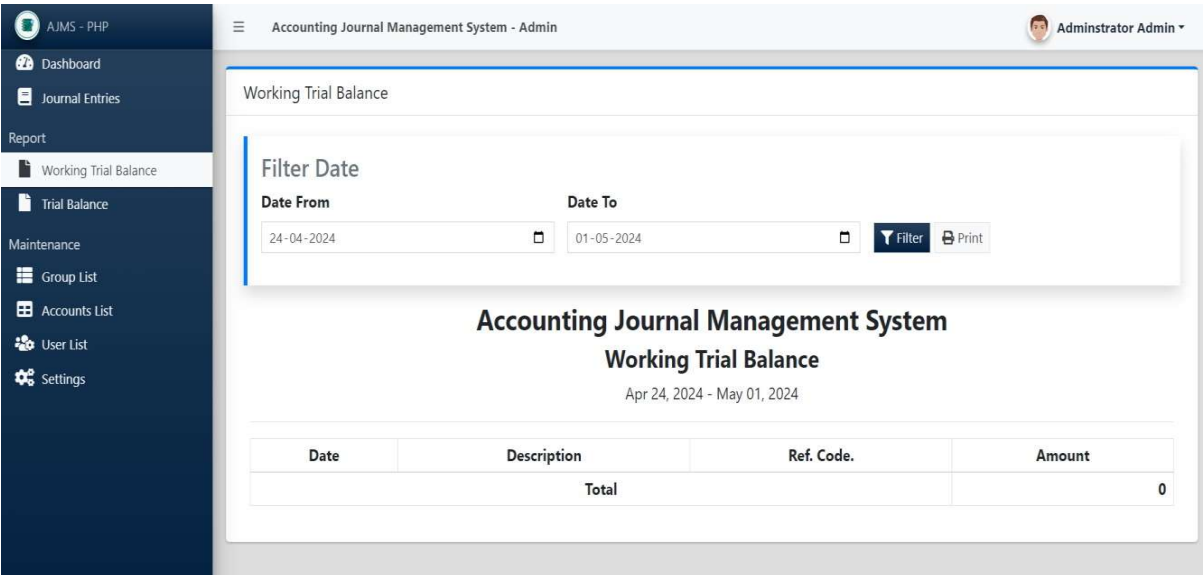


Fig (8) register page

Interface



Fig (9) interface

Accounting Journal Management System - Admin

Administrator Admin

Journal Entries

Add New Journal Entry

Show 10 entries

Search:

Date	Journal Code	Description	Debit	Credit	Encoded By	Action
Feb 01, 2022	202202-00001	Capital Cash Owner's Capital	15,000	15,000	admin	Action
Feb 01, 2022	202202-00002	Sample Office salaries expense Office Salaries Payable	5,000	5,000	admin	Action
Feb 01, 2022	202202-00003	Sample 102 Commissions earned Commissions earned Accounts Receivable	8,000	5,000 3,000	admin	Action

Showing 1 to 3 of 3 entries

Previous 1 Next

Fig (10) journal entries

Fig (11) account list

Accounting Journal Management System - Admin

Administrator Admin

Show 10 entries

Search:

#	Date Created	Account	Description	Status	Action
1	2022-02-01 10:11	Accounts Payable	Accounts Payable	Active	Action
2	2022-02-01 10:10	Accounts Receivable	Accounts Receivable	Active	Action
3	2022-02-01 11:42	Amortization expense	Amortization expense	Active	Action
4	2022-02-01 10:09	Cash	Cash	Active	Action
5	2022-02-01 10:15	Cash dividends	Cash dividends	Active	Action
6	2022-02-01 10:09	Cash Equivalents	Cash Equivalents	Active	Action
7	2022-02-01 10:18	Commissions earned	Commissions earned	Active	Action
8	2022-02-01 10:14	Common stock dividend distributable	Common stock dividend...	Active	Action
9	2022-02-01 10:14	Common stock, no par value	Common stock, no par value	Active	Action
10	2022-02-01 10:14	Common Stock, Par Value	Common Stock, Par Value	Active	Action

Showing 1 to 10 of 48 entries

Previous 1 2 3 4 5 Next

4.2 FEATURES

- An online journal management system typically includes the following features:
- 1. **Submission Management**: Authors can submit their manuscripts online, including uploading files, entering metadata, and tracking the status of their submissions.
- 2. **Peer Review System**: Editors can assign reviewers, manage the review process, and make decisions based on reviewer feedback.
- 3. **Editorial Workflow**: Editors can manage the entire editorial process, from submission to publication, including assigning reviewers, making decisions, and communicating with authors.
- 4. **Publication Management**: Editors can schedule accepted manuscripts for publication, manage article versions, and publish articles online.
- 5. **User Management**: Administrators can manage user roles and permissions, including authors, reviewers, and editors.
- 6. **Content Management**: The system provides tools for managing and organizing published content, including articles, issues, and supplementary materials.
- 7. **Indexing and Search**: The system includes indexing features to make published content searchable and discoverable.

- 8. ***Analytics and Reporting***: The system provides analytics and reporting tools to track submission and publication metrics, such as acceptance rates and publication timelines.
- 9. ***Integration with Other Systems***: The system can integrate with other systems, such as plagiarism detection tools, to enhance the editorial process.
- 10. ***Customization and Flexibility***: The system is often customizable to meet the specific needs of a journal, including branding, workflows, and submission guidelines.

These features help streamline the editorial process, improve efficiency, and enhance the overall management of scholarly journals.

4.2.1 Novelty of the proposal

The proposed online journal management system (OJMS) represents a significant advancement in the field of scholarly publishing, offering a range of novel features and functionalities that distinguish it from existing systems. This essay will explore the key aspects of novelty in the proposed OJMS, including its integration of blockchain technology, the use of AI algorithms for manuscript screening and review assignment, and its advanced analytics and reporting tools.

One of the primary innovations of the proposed OJMS is its integration of blockchain technology. Blockchain is a decentralized, distributed ledger technology that offers several advantages for journal management systems, including enhanced security, transparency, and immutability. By leveraging blockchain, the OJMS ensures that all actions within the system, such as manuscript submissions, peer reviews, and editorial decisions, are recorded in a secure and tamper-proof manner. This not only enhances

the overall integrity of the journal's processes but also provides authors, reviewers, and editors with greater confidence in the system's transparency and fairness.

Furthermore, the use of blockchain technology can help address some of the key challenges facing traditional journal management systems, such as issues of trust, accountability, and data integrity. For example, by storing manuscript metadata and review histories on the blockchain, the OJMS can ensure that this information is secure and cannot be altered or tampered with.

This can help prevent issues such as reviewer bias or manipulation of review outcomes, thereby enhancing the credibility and reliability of the journal's peer review process.

Another key innovation of the proposed OJMS is its use of AI algorithms for manuscript screening and review assignment. AI has the potential to revolutionize the peer review process by enabling more efficient and effective matching of manuscripts with appropriate reviewers. By analyzing the content of submitted manuscripts and the expertise of potential reviewers, AI algorithms can quickly and accurately identify suitable reviewers, thereby speeding up the review process and improving the quality of reviews.

In addition to manuscript screening and review assignment, AI can also be used to automate other aspects of the editorial process, such as plagiarism detection, language editing, and formatting. This can help streamline the editorial workflow and reduce the burden on editors and reviewers, allowing them to focus on more value-added tasks

CHAPTER 5

RESULTS AND 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

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

Key features:

1. ***Blockchain Integration***: OJMS integrates blockchain technology to enhance the security, transparency, and immutability of the journal's data and transactions. By leveraging blockchain, OJMS ensures that all actions within the system, such as manuscript submissions, peer reviews, and editorial decisions, are recorded in a secure and tamper-proof manner.

2. ***AI Algorithms***: OJMS uses AI algorithms for manuscript screening and review assignment. These algorithms analyze the content of submitted manuscripts and the expertise of potential reviewers to quickly and accurately match manuscripts with suitable reviewers, thereby speeding up the review process and improving the quality of reviews.

3. ***Advanced Analytics and Reporting Tools***: OJMS provides editors and administrators with advanced analytics and reporting tools that offer valuable insights into the journal's performance, submission trends, and reviewer behaviors. These tools enable data-driven decision-making, which is crucial for enhancing the efficiency and effectiveness of the journal's operations.

4. ***Efficient Editorial Workflow***: OJMS streamlines the editorial workflow by automating various tasks such as manuscript submission, peer review, and editorial decision-making. This reduces the time and effort required to manage a journal, leading to faster turnaround times for authors and more efficient communication between authors, reviewers, and editors.

PROGRAM :

```
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-3
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 mb-5">
9
10          <!-- card header -->
11
12          <!-- card body -->
13
14          <div class="card-body pb-3 ps-4 pe-5">
15
16            <div class="cdhd text-center">
17              Add Post
18            </div>
19
20            <form action="" #addpst="ngForm" (submit)="doAddPost(addpst)">
21
22              <!-- project title -->
23
24              <div class="form-group">
25
26                <div class="mb-5">
27                  <div class="input-group ip1">
28                    <div class="input-group-prepend sym1">
29                      <span class="input-group-text ugm">Title </span>
30                    <input type="text" class="form-control" #title />
31                  </div>
32                </div>
33              </div>
34            </form>
35          </div>
36        </div>
37      </div>
38    </div>
39  </div>
40</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-3
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 mb-5">
9
10          <!-- card header -->
11
12          <!-- card body -->
13
14          <div class="card-body pb-3 ps-4 pe-5">
15
16            <div class="cdhd text-center">
17              Add Post
18            </div>
19
20            <form action="" #addpst="ngForm" (submit)="doAddPost(addpst)">
21
22              <!-- project title -->
23
24              <div class="form-group">
25
26                <div class="mb-5">
27                  <div class="input-group ip1">
28                    <div class="input-group-prepend sym1">
29                      <span class="input-group-text ugm">Title </span>
30                    <input type="text" class="form-control" #title />
31                  </div>
32                </div>
33              </div>
34            </form>
35          </div>
36        </div>
37      </div>
38    </div>
39  </div>
40</div>
```

```

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
14 doSearch(search:string){
15   this.usrSer.searchUsers(search).subscribe((data:any[])=>{
16     console.log(data);
17     this.posts1=data;
18   },(error:any)=>{
19     console.log(error);
20   })
21 }
22
23 }
24
25
26
27
28

```

```

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 = "somenthing went wrong";
30
31         });
32         formdata.reset( )
33

```

```

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

```

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 titile -->
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 unmi"> Tittle </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="pdesc" id="pdesc" 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">Decription</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 unnm"> Author </span>
47             </div>
48
49             <input type="text" class="form-control" name="pauthor" id="pauthor" aria-describedby="h
50
51         </div>
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 unnm22 p-2 d-grid text-center">Content</label>
60             <textarea class="form-control txtar" name="mncnt" id="mncnt" rows="8" [ngModel]="post
61
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>

```

```

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     });
34   }
35 }

```

```

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 }

```

CONCLUSION

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. The online journal management system (OJMS) represents a transformative innovation in the field of scholarly publishing, offering a wide range of benefits and opportunities for journals, authors, reviewers, and readers. In this conclusion, we will summarize the key advantages of OJMS, discuss its implications for the future of scholarly publishing, and explore potential challenges and limitations. One of the primary advantages of OJMS is its ability to enhance the efficiency and effectiveness of the editorial process. By automating various tasks such as manuscript submission, peer review, and editorial decision-making, OJMS can significantly reduce the time and effort required to manage a journal. For authors, this means faster turnaround times and more transparent communication with editors and reviewers. For reviewers, it means more efficient matching with manuscripts that align with their expertise, leading to higher-quality reviews. And for editors, it means better oversight of the entire editorial workflow, with advanced analytics and reporting tools providing valuable insights into the journal's performance and submission trends.

REFERENCES

- [1] Thomas A Powell, The Complete Reference HTML & CSS, McGraw th HillEducation, 5 Edition July - 2023
- [2] Charles Severance, Python For EveryBody, Shroff Publications July - 2022 .
- [3] Ayush Mauryavanshi, A Complete Overview on: Web Development ST Notion Press 1 Edition 29 -July-2023 . nd
- [4] Eric Mathes, Python Crash Course, 2 Edition, 2023
- [5] R.W. Boyatzis, Transforming qualitative information: Thematic analysis and code development, Thousand Oaks, London, & New Delhi,, SAGE Publications, 1998.
- [6] E. Freeman, B. Bates, K. Sierra, and E. Robson, Head First Design Patterns, O'Reilly Media, 2022.
- [7] A. Fox and D. Patterson, Engineering Software as a Service: An Agile Approach Using Cloud Computing, Strawberry Canyon LLC, 2nd Edition,
- [8] Bern J., Meinel C.: One step further: Extending Electronic Submission into the Reviewing Process, SIGDOC '99
- [9] Journal of Computing and Information Technology, University of Computing Centre, Zagreb.

A Research paper

Design and Implementation of an Online Journal Publication Management System

Veera manoj Gummadi
Dept of computer science and
Engineering (Blockchain Technology)
Sathyabama Institute of science and
technology
Chennai,India.
gummadiveeramanoj@gmail.com

Varun Kumar Battu
Dept of computer science and
Engineering (Blockchain Technology)
Sathyabama Institute of science and
technology
Chennai,India
yvarunkumar4@gmail.com

Lekshmi S Raveendran
Dept of computer science and
Engineering (Assistant professor)
Sathyabama Institute of
science and technology
Chennai,India
lekshmiraveendran3546@gmail.com

Abstract This paper presents the design and implementation of an Online Journal Publication Management System (OJPMS) aimed at streamlining the process of managing scholarly journals, from submission to publication. With the exponential growth of academic research and the increasing demand for efficient peer-review processes, there is a critical need for robust systems to manage the publication workflow. OJPMS is designed to address these challenges by providing a comprehensive platform for authors, editors, and reviewers to collaborate seamlessly in the publication process. The system incorporates features such as manuscript submission, peer-review management, editorial workflow tracking, and publication scheduling. Through a user friendly interface and customizable configurations, OJPMS offers an efficient solution to enhance the quality and efficiency of journal publication.

Key words : Online Journal Publication, Management System, Peer-Review, Manuscript Submission, Editorial Workflow

1. INTRODUCTION

The proliferation of academic research across various disciplines has led to a surge in the number of scholarly journals worldwide. As a result, managing the publication workflow has become increasingly complex, requiring efficient systems to handle the submission, review, and publication processes. Traditional paper-based systems are no longer sufficient to the demands of modern academic publishing, necessitating the development of online solutions.

II .Literature Survey

A comprehensive platform for managing research projects and sharing data. Zenodo: A general-purpose repository for research outputs, including project journals. Figshare: Another repository for various research outputs, with features for collaboration and version control. Several existing platforms offer online project journal publication functionalities.[1]

The Open Science Framework (OSF) provides a comprehensive suite of tools for managing research projects and sharing data, including project journals. Zenodo serves as a general-purpose repository for research outputs, where project journals can be deposited. Figshare offers another platform for research outputs, with features that support collaboration and version control for project journals.[2]

Journals Publications serve a range of purposes. A journal is meant to 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.[3]

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.[4]

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. [5]

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.[6]

III. RELATED WORK

Several online journal management systems exist, each offering different features and functionalities. Platforms such as Editorial Manager, ScholarOne Manuscripts, and Open Journal Systems (OJS) have gained popularity among publishers and academic institutions. However, these systems often come with limitations such as high costs, complex interfaces, or lack of customization options. In response to these challenges, there is a need for a more accessible and adaptable solution tailored to the specific requirements of journal publishers.

IV. SYSTEM ARCHITECTURE

The Online Journal Publication Management System (OJPMS) is designed as a web-based application, allowing users to access the platform from any internet-enabled device. The system architecture comprises three main components: the frontend interface, backend server, and database. The frontend interface provides a user-friendly environment for authors, editors, and reviewers to interact with the system. The backend server handles the core logic of the application, including manuscript management, user authentication, and workflow automation. The database stores all relevant data, including user profiles, manuscript submissions, review comments, and publication records.

V. FEATURES AND FUNCTIONALITIES OJPMS

offers a range of features to support the end-to-end publication workflow:

1. **Manuscript Submission:** Authors can submit their manuscripts online, along with metadata such as title, abstract, and keywords. The system supports various file formats and allows authors to track the status of their submissions.
2. **Peer-Review Management:** Editors can assign reviewers to evaluate submitted manuscripts and manage the peer-review process efficiently. Reviewers can submit their comments and recommendations through the system, facilitating transparent communication between authors, editors, and reviewers.
3. **Editorial Workflow Tracking:** The system provides editors with tools to monitor the progress of manuscripts through the editorial workflow. Editors can track submissions, assign reviewers, make editorial decisions, and communicate with authors seamlessly.
4. **Publication Scheduling:** Once a manuscript has been accepted for publication, editors can schedule its release date and manage the publication pipeline. The system generates digital object identifiers (DOIs) and metadata for published articles, ensuring proper indexing and discoverability.

A. Implementation Details

OJPMS is developed using modern web technologies such as HTML, CSS, JavaScript, and Node.js for the frontend interface, Express.js for the backend server, and MongoDB for the database. The system is designed to be scalable, secure, and customizable, allowing publishers to adapt it to their specific requirements. Continuous integration and deployment (CI/CD) practices are employed to ensure the reliability and stability of the system.

VI. EVALUATION AND PERFORMANCE

To assess the effectiveness of OJPMS, usability tests and performance evaluations are conducted with stakeholders, including authors, editors, and reviewers. Feedback is collected to identify areas for improvement and refine the system's features and functionalities. Key performance indicators such as submission throughput, review turnaround time, and publication latency are monitored to gauge the system's efficiency and reliability.

VII. CONCLUSION

In conclusion, OJPMS offers a comprehensive solution for managing the publication workflow of scholarly journals. By leveraging modern web technologies and best practices in software engineering, the system provides an efficient and user-friendly platform for authors, editors, and reviewers to collaborate effectively. Future work includes further refinement of the system based on user feedback, integration with external services such as plagiarism detection tools and indexing databases, and expansion to support additional publishing formats such as preprints and data papers.

VIII. ACKNOWLEDGMENTS

We extend our sincerest gratitude to the developers and designers who dedicated countless hours to crafting the intricate framework of our online journal publication management system. Their technical expertise and innovative approach have transformed abstract ideas into a tangible platform that streamlines the process of scholarly communication. We also express our appreciation to the researchers and academics whose valuable feedback and insights have guided the continuous improvement of our system. Their constructive criticism and suggestions have been instrumental in shaping its functionality and usability, ensuring it meets the evolving needs of the academic community. Furthermore, we thank the editorial teams and reviewers who diligently evaluate submissions and uphold the standards of academic excellence. Their commitment to scholarly rigor and integrity strengthens the credibility of our platform and fosters a culture of intellectual discourse.

IX. REFERENCES

- [1] [Online]. Available: <https://www.editorialmanager.com/>
- [2] [Online]. Available: <https://clarivate.com/webofsciencegroup/solutions/scholarone-manuscripts/>
- [3] Open Journal Systems (OJS)
- [4] A. Thomas and Powell, 2022.
- [5] C. Severance, 2022.
- [6] A. Mauryavanshi, 2023
- [7] E. Mathes, 2023

B SOURCE CODE

```
<?php
ob_start();
ini_set('date.timezone','Asia/Manila');
date_default_timezone_set('Asia/Manila');
session_start();

require_once('initialize.php');
require_once('classes/DBConnection.php');
require_once('classes/SystemSettings.php');
$db = new DBConnection;
$conn = $db->conn;

function redirect($url='') {
    if(!empty($url))
        echo '<script>location.href="'.base_url . $url.'"</script>';
}

function validate_image($file) {
    if(!empty($file)) {
        // exit;
        $ex = explode('?', $file);
        $file = $ex[0];
        $param = isset($ex[1]) ? '?' . $ex[1] : '';
        if(is_file(base_app.$file)) {
            return base_url.$file.$param;
        } else {
            return base_url.'dist/img/no-image-available.png';
        }
    } else {
        return base_url.'dist/img/no-image-available.png';
    }
}

function isMobileDevice() {
    $aMobileUA = array(
        '/iphone/i' => 'iPhone',
        '/ipod/i' => 'iPod',
        '/ipad/i' => 'iPad',
        '/android/i' => 'Android',
        '/blackberry/i' => 'BlackBerry',
        '/webos/i' => 'Mobile'
    );

    //Return true if Mobile User Agent is detected
    foreach($aMobileUA as $sMobileKey => $sMobileOS) {
        if(preg_match($sMobileKey, $_SERVER['HTTP_USER_AGENT'])) {
            return true;
        }
    }
}
```

```

        //Otherwise return false..
        return false;
    }
    ob_end_flush();
?>
/**
 * -----
 * AdminLTE Layout.js
 * License MIT
 * -----
 */

import $ from 'jquery'

/**
 * Constants
 * =====
 */

const NAME = 'Layout'
const DATA_KEY = 'lte.layout'
const JQUERY_NO_CONFLICT = $.fn[NAME]

const SELECTOR_HEADER = '.main-header'
const SELECTOR_MAIN_SIDEBAR = '.main-sidebar'
const SELECTOR_SIDEBAR = '.main-sidebar .sidebar'
const SELECTOR_CONTENT = '.content-wrapper'
const SELECTOR_CONTROL_SIDEBAR_CONTENT = '.control-sidebar-content'
const SELECTOR_CONTROL_SIDEBAR_BTN = '[data-widget="control-sidebar"]'
const SELECTOR_FOOTER = '.main-footer'
const SELECTOR_PUSHMENU_BTN = '[data-widget="pushmenu"]'
const SELECTOR_LOGIN_BOX = '.login-box'
const SELECTOR_REGISTER_BOX = '.register-box'

const CLASS_NAME_SIDEBAR_COLLAPSED = 'sidebar-collapse'
const CLASS_NAME_SIDEBAR_FOCUSED = 'sidebar-focused'
const CLASS_NAME_LAYOUT_FIXED = 'layout-fixed'
const CLASS_NAME_CONTROL_SIDEBAR_SLIDE_OPEN = 'control-sidebar-slide-open'
const CLASS_NAME_CONTROL_SIDEBAR_OPEN = 'control-sidebar-open'

const Default = {
    scrollbarTheme: 'os-theme-light',
    scrollbarAutoHide: 'l',
    panelAutoHeight: true,
    panelAutoHeightMode: 'min-height',
    loginRegisterAutoHeight: true
}

```

```

/**
 * Class Definition
 * =====
 */

class Layout {
  constructor(element, config) {
    this._config = config
    this._element = element
  }

  // Public

  fixLayoutHeight(extra = null) {
    const $body = $('body')
    let controlSidebar = 0

    if ($body.hasClass(CLASS_NAME_CONTROL_SIDEBAR_SLIDE_OPEN) ||
$body.hasClass(CLASS_NAME_CONTROL_SIDEBAR_OPEN) || extra ===
'control_sidebar') {
      controlSidebar = $(SELECTOR_CONTROL_SIDEBAR_CONTENT).outerHeight()
    }

    const heights = {
      window: $(window).height(),
      header: $(SELECTOR_HEADER).length > 0 ? $(SELECTOR_HEADER).outerHeight()
: 0,
      footer: $(SELECTOR_FOOTER).length > 0 ? $(SELECTOR_FOOTER).outerHeight()
: 0,
      sidebar: $(SELECTOR_SIDEBAR).length > 0 ? $(SELECTOR_SIDEBAR).height() :
0,
      controlSidebar
    }

    const max = this._max(heights)
    let offset = this._config.panelAutoHeight

    if (offset === true) {
      offset = 0
    }

    const $contentSelector = $(SELECTOR_CONTENT)

    if (offset !== false) {
      if (max === heights.controlSidebar) {
        $contentSelector.css(this._config.panelAutoHeightMode, (max + offset))
      } else if (max === heights.window) {

```

```

        $contentSelector.css(this._config.panelAutoHeightMode, (max + offset)
- heights.header - heights.footer)
    } else {
        $contentSelector.css(this._config.panelAutoHeightMode, (max + offset)
- heights.header)
    }

    if (this._isFooterFixed()) {
        $contentSelector.css(this._config.panelAutoHeightMode,
parseFloat($contentSelector.css(this._config.panelAutoHeightMode)) +
heights.footer)
    }
}

if (!$body.hasClass(CLASS_NAME_LAYOUT_FIXED)) {
    return
}

if (typeof $.fn.overlayScrollbars !== 'undefined') {
    $(SELECTOR_SIDEBAR).overlayScrollbars({
        className: this._config.scrollbarTheme,
        sizeAutoCapable: true,
        scrollbars: {
            autoHide: this._config.scrollbarAutoHide,
            clickScrolling: true
        }
    })
} else {
    $(SELECTOR_SIDEBAR).css('overflow-y', 'auto')
}
}

fixLoginRegisterHeight() {
    const $body = $('body')
    const $selector = $('`${SELECTOR_LOGIN_BOX}, ${SELECTOR_REGISTER_BOX}`')

    if ($selector.length === 0) {
        $body.css('height', 'auto')
        $('html').css('height', 'auto')
    } else {
        const boxHeight = $selector.height()

        if ($body.css(this._config.panelAutoHeightMode) !== boxHeight) {
            $body.css(this._config.panelAutoHeightMode, boxHeight)
        }
    }
}
}

```

```

// Private

_init() {
  // Activate layout height watcher
  this.fixLayoutHeight()

  if (this._config.loginRegisterAutoHeight === true) {
    this.fixLoginRegisterHeight()
  } else if (this._config.loginRegisterAutoHeight ===
parseInt(this._config.loginRegisterAutoHeight, 10)) {
    setInterval(this.fixLoginRegisterHeight,
this._config.loginRegisterAutoHeight)
  }

  $(SELECTOR_SIDEBAR)
    .on('collapsed.lte.treeview expanded.lte.treeview', () => {
      this.fixLayoutHeight()
    })

  $(SELECTOR_MAIN_SIDEBAR)
    .on('mouseenter mouseleave', () => {
      if ($('#body').hasClass(CLASS_NAME_SIDEBAR_COLLAPSED)) {
        this.fixLayoutHeight()
      }
    })

  $(SELECTOR_PUSHMENU_BTN)
    .on('collapsed.lte.pushmenu shown.lte.pushmenu', () => {
      setTimeout(() => {
        this.fixLayoutHeight()
      }, 300)
    })

  $(SELECTOR_CONTROL_SIDEBAR_BTN)
    .on('collapsed.lte.controlsidebar', () => {
      this.fixLayoutHeight()
    })
    .on('expanded.lte.controlsidebar', () => {
      this.fixLayoutHeight('control_sidebar')
    })

  $(window).resize(() => {
    this.fixLayoutHeight()
  })

  setTimeout(() => {
    $('#body.hold-transition').removeClass('hold-transition')
  }, 50)

```

```

    }

    _max(numbers) {
      // Calculate the maximum number in a list
      let max = 0

      Object.keys(numbers).forEach(key => {
        if (numbers[key] > max) {
          max = numbers[key]
        }
      })

      return max
    }

    _isFooterFixed() {
      return $(SELECTOR_FOOTER).css('position') === 'fixed'
    }

    // Static

    static _jQueryInterface(config = '') {
      return this.each(function () {
        let data = $(this).data(DATA_KEY)
        const _options = $.extend({}, Default, $(this).data())

        if (!data) {
          data = new Layout($(this), _options)
          $(this).data(DATA_KEY, data)
        }

        if (config === 'init' || config === '') {
          data._init()
        } else if (config === 'fixLayoutHeight' || config ===
'fixLoginRegisterHeight') {
          data[config]()
        }
      })
    }
  }
}

/**
 * Data API
 * =====
 */

$(window).on('load', () => {
  Layout._jQueryInterface.call($('body'))
})

```

```

})

$(`${SELECTOR_SIDEBAR} a`)
  .on('focusin', () => {
    $(SELECTOR_MAIN_SIDEBAR).addClass(CLASS_NAME_SIDEBAR_FOCUSED)
  })
  .on('focusout', () => {
    $(SELECTOR_MAIN_SIDEBAR).removeClass(CLASS_NAME_SIDEBAR_FOCUSED)
  })

/**
 * jQuery API
 * =====
 */

$.fn[NAME] = Layout._jQueryInterface
$.fn[NAME].Constructor = Layout
$.fn[NAME].noConflict = function () {
  $.fn[NAME] = JQUERY_NO_CONFLICT
  return Layout._jQueryInterface
}

export default Layout

```