

# **SCHOLARAID**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE  
AWARD OF DEGREE OF

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE**



Submitted by  
Harshit Mishra (2100290120085)

Jatish Kumar (2100290120089)

Deepanshu (2100290120068)

Ibtesam Ansari (2100290120087)

**Supervised by**  
**Prof. Pawan Kumar Pal**  
Project Supervisor  
Session 2024-25

**DEPARTMENT OF COMPUTER SCIENCE**

**KIET GROUP OF INSTITUTIONS, GHAZIABAD**

(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)

**May 2025**

## **DECLARATION**

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

Signature

Name:- Harshit Mishra

Roll No.:- 2100290120085

Date:- 12-05-2025

Signature

Name:-Deepanshu

Roll No.:- 2100290120068

Date:-12-05-2025

Signature

Name:- Jatish Kumar

Roll No.:- 2100290120089

Date:- 12-05-2025

Signature

Name:-Ibtesam Umar Ansari

Roll No.:- 2100290120087

Date:-12-05-2025

## **CERTIFICATE**

This is to certify that Project Report entitled “Scholar Aid ” which is submitted by in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Date:12-05-2025**

**Supervisor**

Name: Prof. Pawan Kumar Pal

Designation:Project Supervisor

## **ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the project undertaken during B. Tech Final Year. We owe special debt of gratitude to Prof. Pawan Kumar Pal, Department of Computer Science, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Srivastava, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We are also thankful to our friends for their contribution in completion of the project.

Date :

Signature

Name:- Harshit Mishra

Roll No.:- 2100290120085

Signature

Name:-Deepanshu

Roll No.:- 2100290120068

Signature

Name:- Jatish Kumar

Roll No.:- 2100290120089

Signature

Name:-Ibtesam Umar Ansari

Roll No.:- 2100290120087

## **ABSTRACT**

In today's situations, access to education is paramount for personal and societal advancement. However, the financial constraints often pose a significant barrier for many aspiring students. To address this challenge and promote educational equality, the development of a comprehensive portal for accessing information on national and international scholarships becomes necessary. This describes about the reason and the conceptualization of such a portal, which aims to serve as a centralized platform for individuals seeking various scholarship opportunities. The portal will provide information on scholarships offered by government, non-government and private organizations at the national and international stages.

Key features of the portal include:

1. Realtime Database: A vast database containing detailed information about available scholarships, including eligibility criteria, application deadlines.
2. Filtering Results: Users can efficiently search and filter scholarships based upon criteria such as academic stream, level of education, geographical location, and other specific eligibility requirements.
3. User Profile: Personalized user profiles allows the individuals to save preferred scholarship opportunities, track application status, and receive recommendations based on their academic background and interests.
4. Resource Hub: A good collection of resources including tips for crafting matched scholarship applications, provide guidance in writing personal statements, and links for additional funding sources and support services.
5. Discussion Forum: Integration of discussion areas and networking features to facilitate peer-to-peer support, mentor.

## TABLE OF CONTENTS

DECLARATION.....	ii
CERTIFICATE.....	iii
ACKNOWLEDGEMENTS.....	iv
ABSTRACT.....	v
LIST OF FIGURES.....	viii
LIST OF TABLES.....	ix
LIST OF ABBREVIATIONS.....	x
SDG MAPPING WITH JUSTIFICATION .....	
CHAPTER 1 INTRODUCTION	1-3
1.1    Introduction to Project	1
1.2    Project Category	3
1.3    Objectives	3
CHAPTER 2 LITERATURE REVIEW	4-8
2.1    Literature Review	4
2.2    Problem Formulation	8
CHAPTER 3 PROPOSED SYSTEM	9-11
3.1    Modules	9
3.2    Unique Features of The System	10
CHAPTER 4 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION	12-19
4.1    Feasibility Study (Technical, Economical, Operational)	12
4.2    Software Requirement Specification	13
4.2.1    Data Requirement	13
4.2.2    Functional Requirement	14
4.2.3    Performance Requirement	14
4.2.4    Maintainability Requirement	14

4.2.5	Security Requirement	15
4.3	SDLC Model Used	15
4.4	System Design	16
4.4.1	Data Flow Diagrams	16
4.4.2	Use Case Diagrams	17
4.5	Database Design	18
<b>CHAPTER 5 IMPLEMENTATION</b>		<b>20-26</b>
5.1	Introduction Tools and Technologies Used	20
5.2	Sample Code	20
<b>CHAPTER 6 TESTING, AND MAINTENANCE</b>		<b>27</b>
6.1	Testing Techniques Used	27
<b>CHAPTER 7 RESULTS AND DISCUSSIONS</b>		<b>28-52</b>
7.1	Description of Modules with Snapshots	28
7.2	Technology Discussions	32
<b>CHAPTER 8 CONCLUSION AND FUTURE SCOPE</b>		<b>53-56</b>
Conclusion		53
Future Scope		53
Research Paper Acceptance Proof		55
Proof of patent publication (Screenshot of Publication)		56
<b>REFERENCES (Only in IEEE FORMAT)</b>		<b>58</b>

## **LIST OF FIGURES**

<b>Figure No.</b>	<b>Description</b>	<b>Page No.</b>
1.1	Categorization of routing protocol	2
2.1	Three models	9
3.1	Resolving failed links in DSDV	12

## **LIST OF TABLES**

<b>Table. No.</b>	<b>Description</b>	<b>Page No.</b>
1.1	General Simulation parameters	22

## **LIST OF ABBREVIATIONS**

NAM	Network Animator
MANET	Mobile Ad-hoc Network
DSR	Dynamic Sequence Resource
AODV	Ad-hoc On-Demand Vector
OSI	Open System Interconnections
TCP/IP	Transmission Control Protocol/Internet Protocol
Pdf	Packet Drop Fraction
GSR	Global State Routing

## **SDG Mapping with Justification**

### **1. SDG 4: (Quality Education)**

The portal promotes inclusive and equitable access to education by helping students, especially those from marginal backgrounds ,find and apply for scholarships, decreasing financial barriers to learning.

### **2. SDG 5: (Gender Equality)**

Many scholarships are targeted specifically for female students. The portal highlights these opportunities, encouraging female enrollment and supporting women's empowerment through education.

### **3. SDG 10: (Reduced Inequalities)**

The portal offers scholarships tailored for under-privileged groups like SC/ST, OBC, minority communities, promoting equal opportunity and reducing social and economic disparities.

### **4. SDG 16: (Peace, Justice and Strong Institutions)**

The inclusion of transparent tracking, grievance handling, and proper verification mechanism ensures accountability, fairness, and trust in public services.

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Introduction**

A scholarship is a type of payment awarded to assist a student in their studies and is based not just on great academics but other influencing contributions. Receiving a scholarship is a massive achievement for an individual who is either studying in a high school developing new skills or embarking upon their career. Unlike a student loan, there is no payback for the scholarship money. The young practitioner now has a well-structured financial base from which to launch their career. Scholarships can provide access to elite institutions of learning for enhanced career opportunities, giving scholars a golden opportunity to be prepared for a better future.

There has been Rs. 94,800 crores and Rs. 99,300 crores allocated by way of budgetary support to the education sector by the Indian government in 2020-21, and the public education system as a whole remains appalling. Also, especially in the public school and college sector, which predominantly serves the vulnerable under-privileged student population, the students are receiving low-quality education. The students are left to decide their vocation and job opportunities in the informal sector in order to earn, sooner, rather than later. The situation is exacerbated by the fact that the dropout-rate is excessively high, majorly in class 11 and class 12 of the intermediate stage.

The scholarship has become important because of the following reasons:

1. Increasing cost: The increase in the cost of college, school, and university fees is the number one reason scholarships are helpful for a student. Every year, there is a minimum 5% increase in the tuition fee, particularly for international students.

Therefore, it is highly advantageous to enrol in programs that are fully funded and keep a student out of all the financial stress.

2.Student Loan Rate is shooting up: Through controlled studies, many urban families are beginning to get mortgages and loans. In an attempt to send their students to university for higher education. Additionally, the loan rates in banks rise every year and this is why scholarships matter.

3.Basic Living Expenses: College and university students also must consider costs for food and travel. These costs can be gradual and can create a significant amount of harassment. It is even more daunting when the student and their family must pay for the educational tuition cost, and have already spent too much. In contrast, receiving a scholarship with grants that care for all these associated costs can be very beneficial.

4.Study Abroad: There is no better approach for a student to achieve their dream to study abroad than through scholarships. Applying to a highly regarded and reputable program also assists one's academic opportunities and personal development. This provides a way for students to receive exposure to a global context and build their intercultural competence. Scholarships from more established institutions will generally pay for all costs associated with being a student.

Our Scholarship Portal has a clean and elegant user interface that will help students discover and learn about scholarships. Students will be able to search for scholarships and access scholarship details such as the application procedure, deadlines, and the list of required documents. Students are also equipped with eligibility criteria filters by providing their student background information like age, gender, family income, and many more fields, which gives them a list of scholarships they can apply for all in one place.

## **1.2 Project Category**

Smart Education - The Smart Education project category is focused on using technology to improve learning experiences, use digital classrooms, tailor interactive content, and ensure equity, efficiency, and personalization in addressing the needs of students and teachers.

## **1.3 Objectives**

**Objective 1:** The purpose of this project is to create a web portal that allows students to fill the correct information about their eligibility criteria , so the students will see only those scholarships which are according to their eligibility criteria. Therefore, it will help student to see all scholarships at one place and they do not have to anywhere to search scholarships.

**Objective 2:** The purpose is to provide a medium for the organizations to publish their scholarships, and benefit from providing scholarships to selected students, therefore providing benefits to students from the organizations. An administrator will be responsible for oversight of the organizations, and their scholarships.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Literature Review

##### **The Impact of Scholarships on Students' Academic Performance: A Case of Tertiary Institutions in Enugu State, Nigeria**

Scholarships are assigned for many different reasons and almost always tied to what the individual or group granting the money, is trying to accomplish. With this said, in an effort to bring knowledge to the matter of how scholarships affect student grades in universities and colleges in Enugu State, Nigeria, we surveyed 540 students across 12 institutions. We used propensity score matching as our method of analysis and determined student scholarship recipients had significantly better grades than non-recipients. Our results indicated that in a scenario without any scholarships, grades could fall 13% overall around. Whereas, if scholarships replaced every student loan, the grades for those students could increase 87% approximately.

To support more college students, the government, non-profits, and others with lots of extra money should give scholarships to students who are having financial problems. This means helping students who don't have enough money for tuition, food, or clothes. The government should work to reinstate public funding to support scholarships to students from the students' home states, regardless if it is a public or private institution.

##### **The Academic Performance of Scholarship Students during Medical School (RBEM)**

It is crucial, not only in Brazil but in many other countries as well, to ensure the success of poor students in higher education, especially for highly demanding

programs like Medicine, for students from low-income families. This study compares the academic performance of medical students on scholarships to the academic performance of medical students not on scholarships. Data consisted of 417 students who graduated from a public University in Brazil from 2010-2013.

We performed some statistical tests to compare the grades of these two groups at mid and end stages of their education, as well as their entrance to residency program scores. Scholarship students had better grades at mid-stage and end-stage of their education than Non-scholarship students; they did not differ scores for entrance to residency programs.

Our findings suggest that scholarships can help students from lower-income backgrounds do better in medical school. We believe that it's important to continue supporting these types of programs to help more students succeed.

### **Do Scholarships Help? Preliminary Results of a Case Study of Students in Scholarship Programmes at Monash University, 1997-2001**

From 1997 to 1999, the Australian government offered scholarships to help students from disadvantaged backgrounds attend college. After this program ended, Monash University started its own scholarship program for these students. This study compares the performance of students who received these scholarships to those who received scholarships based on academic merit and to all other students.

We are looking at how well these students do in college, how many of them stay enrolled, and how many of them graduate. We've also asked students about how the scholarships helped them.

Early results show that the students who received scholarships for being underprivileged students tend to have good grades. While we can't say for sure that only the scholarship money is directly responsible for this, the students we talked to

have said that the scholarships boosted their confidence and made them encouraged about their academic achievements.

## **HOPE or No-HOPE: Merit-Based College Scholarship Status and Financial Behaviours Among College Students**

Merit-based scholarships like “Georgia's HOPE Scholarship”, have become popular in recent years as a path to encourage academic success and spread the access to higher education. These programs helped in boosting enrollment and improve grades, but their long-term effects on the students' financial habits are still being observed and analyzed.

The HOPE Scholarship scheme, started in 1993, offers significant financial help to eligible Georgia students by covering tuition fees, and providing a book allowance. To keep the scholarship, students must maintain at least 3.0 GPA. If they fall below this, they lose the scholarship and may face critical financial challenges.

This study observes the financial habits of former HOPE Scholarship beneficiaries to understand what happens after they lose this support. It examines things like credit card debt, student loans, and financial knowledge to find potential risks and help in guiding policies that could reduce these problems.

With the rising costs of colleges and increasing student loan debt in the U.S., it's important to understand how merit-based scholarships effect the financial conditions of students. By studying those who lost their scholarship, this research helps us better understand how academic performance, financial aid, and student debt are connected.

The findings of this research can provide information to policymakers and higher education institutions in developing strategies to support students who may face financial difficulties after losing their merit-based scholarships. By providing

targeted financial counselling, debt handling programs, and other supporting services, the institutions can help in curbing the negative consequences of losing scholarship eligibility.

Finally, this study aims to contribute to a more comprehensive understanding of the long-term impact of merit-based scholarships on student financial behaviour. By describing the financial challenges faced by former scholarship beneficiaries, policymakers and institutions can work towards creating a more equitable and sustainable higher education system.

### **The China Scholarship Council: An Overview**

In recent years, there has been rapid spread of concern in the U.S. about the Chinese Government's possible influence on Chinese students studying abroad. Some Chinese student groups have reported of feeling pressure from the Chinese government, and the U.S. Department of Justice has accused some Chinese students of visa fraud and playing role as agents for the Chinese Government. One way the Chinese Government might influence students is through their scholarships. The China Scholarship Council (CSC) plays a big role in this by availing funds to thousands of Chinese students to study both in China and the overseas. While the CSC says its goal is to promote international education, it also encourages students receiving scholarships to return to China after they finish their studies.

Here are some key points about the CSC and its impact:

- Significant funds: The CSC funds many students, both foreign and Chinese.
- Focus on graduate students: Most CSC scholarships are awarded to Chinese graduate students.
- Limited Impact on U.S. students: While a smaller percentage of Chinese students in the U.S. receive CSC funding, the actual number is likely to be smaller.
- Encouraging return to China: The CSC strongly encourages scholarship recipients to return to China after their studies are completed.

- Desire to stay in the U.S.: Despite the pressure of returning to China, most Chinese STEM PhD students intend to stay in the U.S.

Understanding the role of CSC and its influence over the Chinese students studying abroad is crucial for observing the complex geopolitical landscape and ensuring the integrity of academic research and innovations.

## 2.2 Problem Formulation

Look, India's packed—people everywhere, you can't miss it. Each year, something wild like 60 million students finish up their studies, all from different backgrounds. Now, we've got this thing called Gross Enrollment Ratio, GER for short. It basically checks how many young adults (18-23 age range, you know the lot) actually get enrolled in higher ed. According to the big wigs at AISHE, for the year 2021-22, our GER sits at 28.4%. Not exactly brag-worthy.

Most school kids just can't swing the cost of college or university, not unless they score a scholarship or bite the bullet and take on some loan. The government dishes out around 82,000 scholarships every year, which honestly, barely scratches the surface. The number of applicants blows that tiny figure out of the water.

Now, on the ground, you've got these NGOs hustling away, trying to promote and fund college for those who need it most. Thing is, most students haven't even heard of them. Same deal with a bunch of companies and their educational funding schemes. Even some religious communities are in on the act, offering financial help under the radar as a part of their social work. But, again, no one really knows.

## CHAPTER 3

### PROPOSED SYSTEM

#### **3.1 Modules**

##### **1.Login Page**

Give a good user interface for all pages. designing is also as important as coding Every field in ., every pages of this app is mandatory. i.e. if any of the field is not filled then a toast message should be displayed saying “PLEASE FILL ALL THE FIELDS”. After logging in, show message “SUCCESSFULLY LOGGEDIN”.

The login page provides two categories of login where the categories are as follows:

- a. Admin : Admin the profile which is allotted to the user of the website who provides or sponsors the scholarship for students on the website.
- b. Student : The student profile which is allotted to the users who are school students, college students, graduates, undergraduates, P.hd students from any field and they can apply to the scholarship through any eligibility criteria.

After login, go to dashboard.

##### **2. Registration page**

Password should be strong. We are adding our account details in this page. Main page will be known as the home page from which the user will be navigating into other subpages of the system.

##### **3.Student**

Student can apply for the scheme using ID card, adhaar card of student and mother, student certificate. Check status of the application , whether it is approved or not.

After applying to the scholarship, it gets saved in the student's record where status can be showed as:

- Applied: This status is shown when the student has just applied on the portal for the scholarship.
- In-Process: This status is shown when the application by the student is completed and moves for further steps
- Approved: This status is shown when the application is approved by the admin or the sponsor.

#### **4.Sponsor**

Sponsor can post the scholarship with the description, eligibility criteria, deadline, e.t.c and proctor the applicants who applied to them, he can accept or reject the application based on his observation.

#### **5.Home**

When we click home button in slide window redirect the page to home page. When we click profile button just show the details of the user. Just add settings button no changes must happen. When we click logout button redirect to login page.

### **3.2 Unique features of the system**

#### **1. Centralized Scholarship Database:**

All scholarships government, private, and institutional are listed in one place. This reduces the effort students need to search multiple sources.

#### **2. User role management:**

The portal supports separate logins for students, administrators, and scholarship providers. Each role gets tailored features and access privileges.

### **3. Track application status:**

Students can track the real-time status of their application—submitted, under review, approved, or rejected. This brings transparency to the entire process.

### **4. Admin dashboard:**

Admins get insights into the number of applications, approvals, rejections, and data helpful for policy planning.

# CHAPTER 4

## REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

### **4.1 Feasibility Study (Technical, Economical, Operational)**

The Scholarship Information Portal aims to provide comprehensive details about national and international scholarships. The platform will serve students, researchers, and professionals seeking financial assistance for education. This study evaluates the feasibility of the project from technical, economic, and operational perspectives.

#### **Technical Feasibility:**

##### **Platform and Technology**

- **Frontend:** React.js or Angular for an interactive and responsive UI.
- **Backend:** Node.js, Django, or Spring Boot for handling requests and data management.
- **Database:** MySQL or MongoDB for structured and unstructured data storage.
- **Hosting:** Cloud-based services such as AWS, Google Cloud, or Azure for scalability.
- **Security:** For ensuring security we implement SSL encryption, OAuth authentication, and data encryption to ensure safe transactions.
- **API Integration:** We can also use Third-party APIs to fetch real-time scholarship details from various institutions.

#### **Development Requirements**

- Skilled developers proficient in full-stack development.
- UI/UX designers for an intuitive interface.
- Cybersecurity experts to ensure secure user data management.

## **Economic Feasibility**

- **Development Costs:** The cost of hiring and facilitating developers, designers, and database admins.
- **Infrastructure Costs:** It includes server hosting, database management, and domain purchase.
- **Marketing Costs:** It includes digital marketing, social media promotions, and partnerships with educational institutions
- **Revenue Generation:** Advertisements from universities, educational platforms, and scholarship sponsors.

## **Operational Feasibility:**

- **Content Management Team:** This management team is required to ensure accurate and updated scholarship information.
- **Support Team:** This team assist users with queries and troubleshooting.
- **Marketing Team:** This team to promotes the portal and engages with potential users.
- Partnership agreements with universities and funding institutions.
- Terms of Service and Privacy Policy to safeguard user interests.

## **4.2 Software Requirement Specification**

### **4.2.1 Data Requirement:**

#### **User Data:**

- User profiles including personal details, academic background, and interests
- Login credentials and authentication tokens
- Application history and saved scholarships

#### **Scholarship Data:**

- Scholarship details which includes its name, description, eligibility criteria, funding amount, and deadlines
- Source information from institutions and funding organizations
- Real-time updates from integrated APIs

#### **Transactional Data:**

- Application submissions and status updates
- Communication logs between applicants and institutions

#### **4.2.2 Functional Requirements**

##### **User Registration and Authentication**

- Users can register via email or social login (OAuth)
- Secure login system with two-factor authentication

##### **Scholarship Search**

- Search scholarships by country, field, funding type, and eligibility
- Advanced filters for personalized recommendations

##### **Scholarship Application Tracking**

- Users can save and track application statuses
- Automated reminders for application deadlines

##### **Scholarship Management for Institutions**

- Institutions can post scholarships and update details
- Review and approve applications

#### **4.2.3 Performance Requirements**

- The system should be able to process and control large datasets efficiently.
- Database queries should be optimized to ensure less latency and fast data retrieval.
- Scalability should be ensured with cloud-based infrastructures to accommodate growing user demand.
- The system should have a good uptime to ensure continuous availability.

#### **4.2.4 Maintainability Requirements**

- **Modular Architecture:** The system should be developed using a microservices for easier updates and maintenance.

- **Coding Documentation:** It should have well-structured and documented codebase to support future development and debugging.
- **Automation Testing:** The unit and integration testing methods should be implemented to catch errors early.
- **Version Control:** For version controlling, Git should be used for source code management with CI/CD pipelines for seamless deployment.
- **Error Logging & Monitoring:** It means Real-time error logging and monitoring tools should be integrated to identify issues proactively.
- **Scalable Infrastructure:** The system should be designed in order to support future enhancements and feature additions with minimal disruption.

#### **4.2.5 Security Requirements**

- **Regular Security Audits:** Periodic security testing and vulnerability assessments should be performed.
- **Data Privacy Compliance:** Must adhere to GDPR and other relevant data protection regulations.
- **Account Activity Monitoring:** Suspicious login attempts should trigger alerts and temporary account lockout mechanisms.

#### **4.3 SDLC Model Used:**

The model adopted by us to develop this project is the Agile model. The Agile model is an iterative and flexible approach to software development that focuses on continuous improvement, collaboration among various departments, and customer feedback. It provides iterative development, customer collaboration, flexibility and frequent releases.

## 4.4 System Design

### 4.4.1 Data flow Diagram

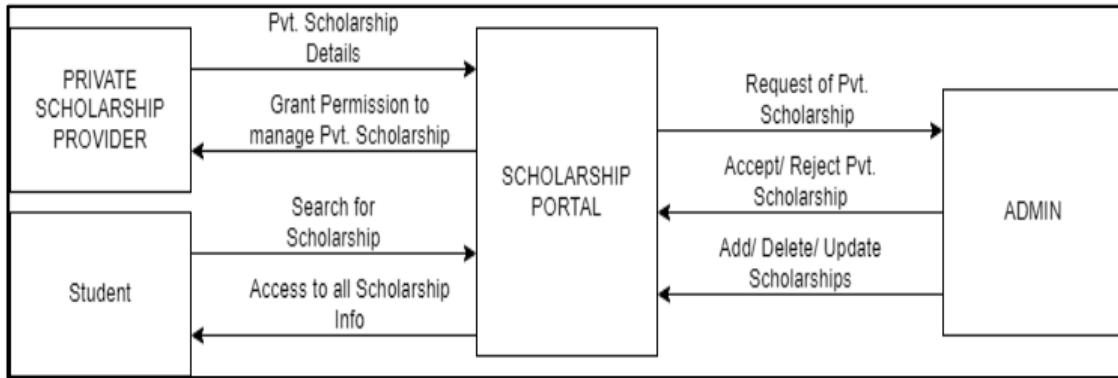


Fig 4.4.1.1

In the above figure 4.4.1.1, this shows a Level-0 Data flow diagram. A level-0 data flow diagram gives an entire overview of the system.

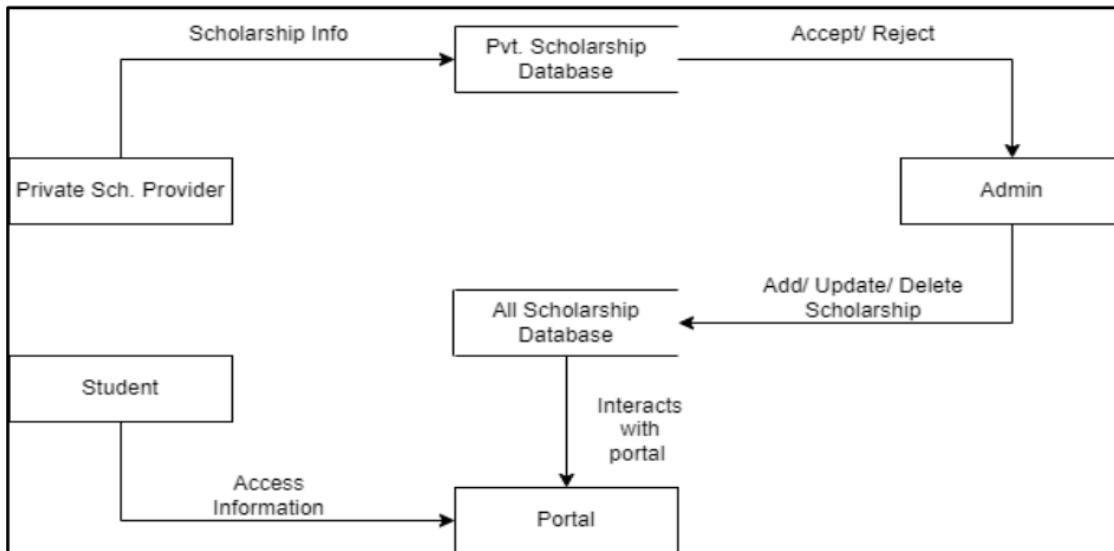


Fig 4.4.1.2

In the above figure 4.4.1.2, this shows a Level-1 Data flow diagram. A level-1 data flow diagram gives a more detailed view of the system, which highlights the main functions of the system.

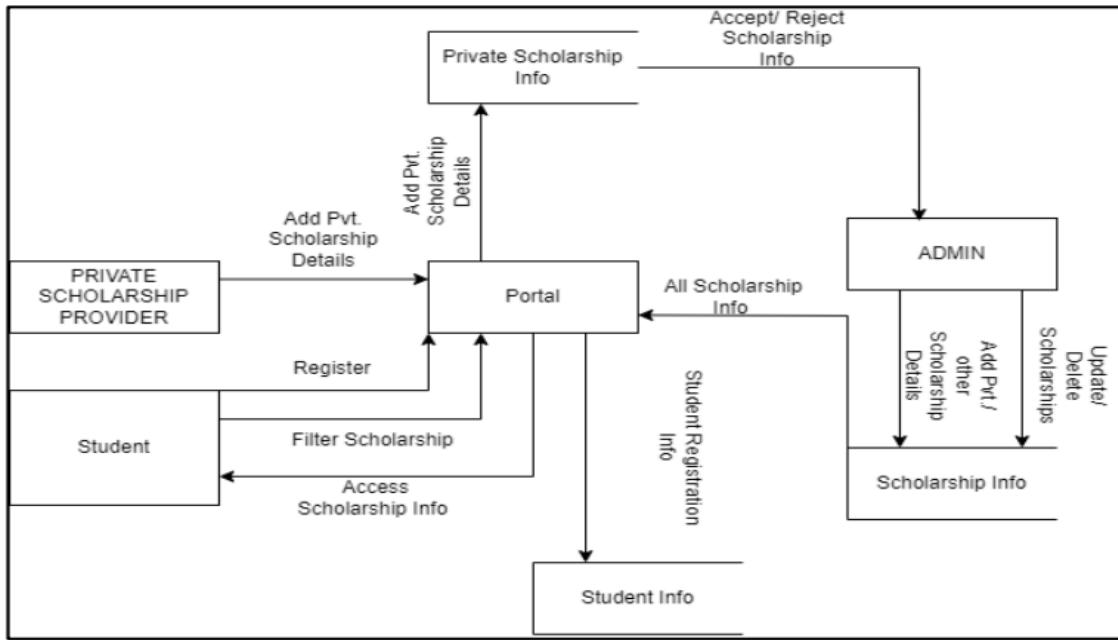


Fig 4.4.1.3

In the above figure 4.4.1.3, this is the level-2 diagram of the system. This level of data flow diagram provides a more detailed view of the system by also describing the subprocesses.

#### 4.4.2 Use Case Diagram

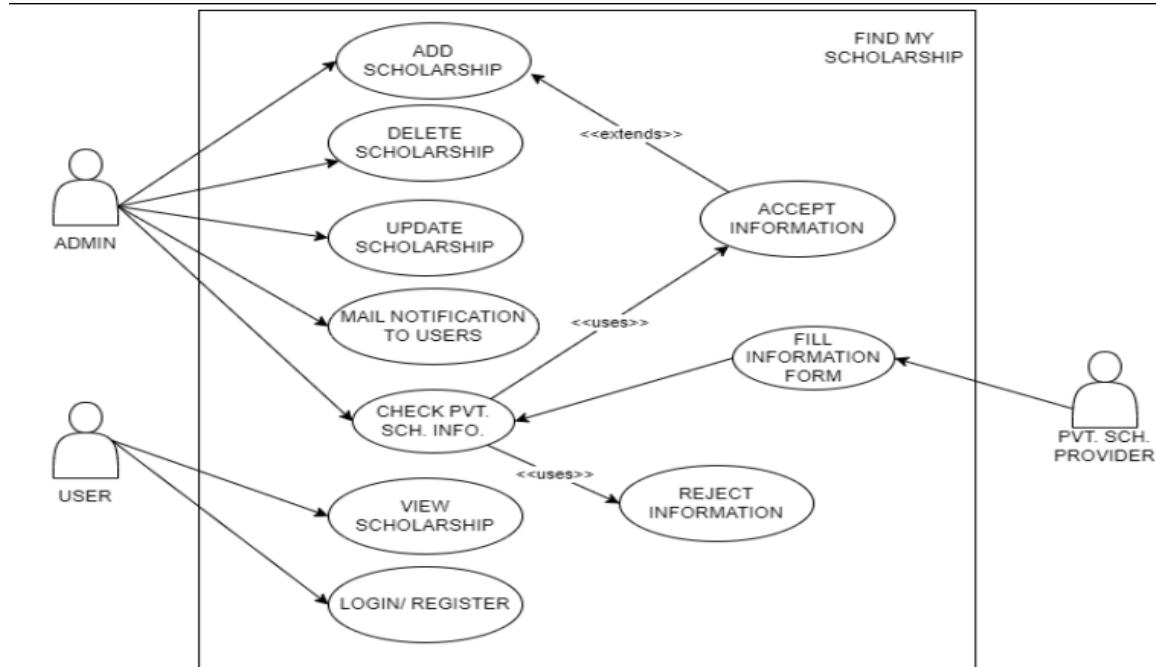


Fig 4.4.2.1

In the above Fig 4.4.2.1, it shows the use case diagram of Scholar Aid portal. It shows the activities held among the Admin, the Student and the external scholarship provider(in case the scholarship is available externally on the sponsor's website).

Functions performed in the systems are as follows:

- login/register by the candidate.
- View available scholarships for students.
- Apply for the scholarship by the students.
- Publish/remove scholarships for students by the admin.
- Update the deadline for the scholarship.
- Approval/rejection of the application by the admin.
- Verification of documents by the admin.

## 4.5 Database Design

### E-R Diagram:

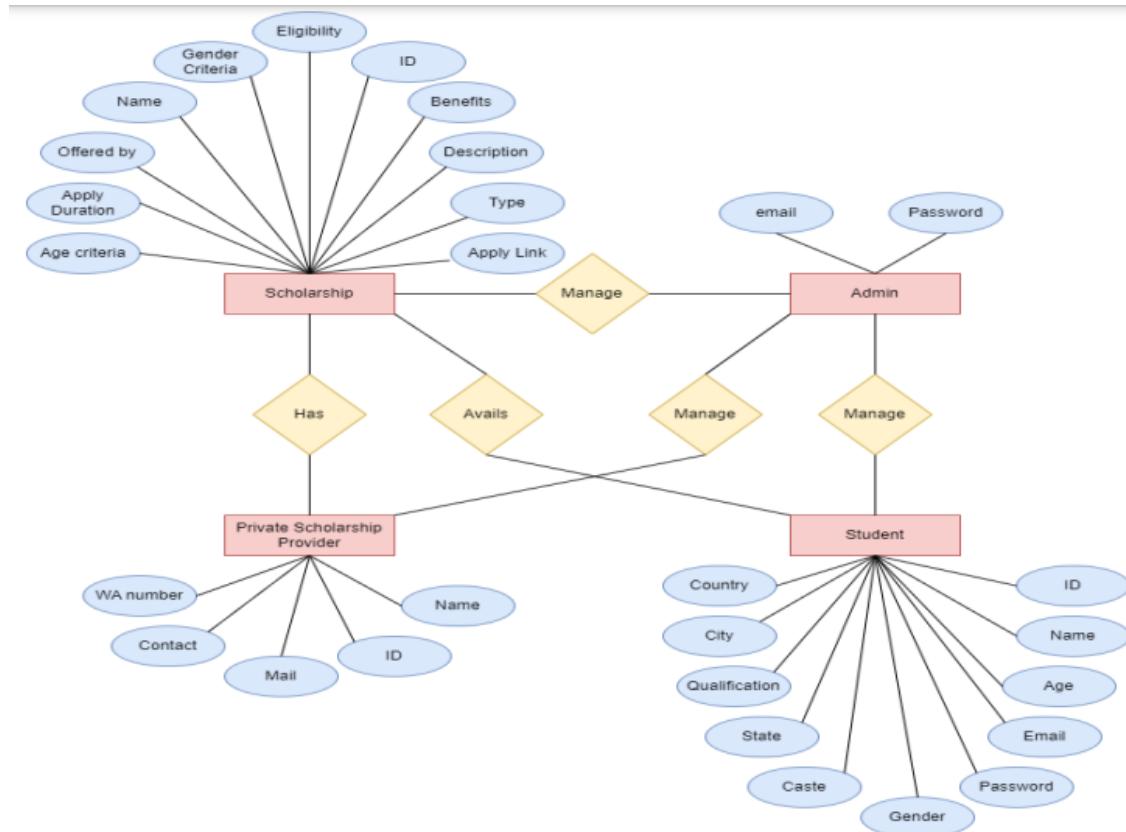


Fig 4.5.1

In the Fig 4.5.1, it shows the ER-diagram of the portal. An ER (Entity-Relationship) Diagram shows the data model of a system by depicting entities, their attributes, and the relationships between them. It helps in understanding how a database is logically organized. ER diagrams are important during the database design process to ensure data is consistent and stored efficiently.

# CHAPTER 5

## IMPLEMENTATION

### 5.1 Tools and Technologies Used:

**Frontend:** HTML, CSS, React JS

**Backend:** Node JS

**Database:** Mongo DB

### 5.2 Sample Code:

**Admin Dashboard:** Figure 5.2.1 shows the implementation code for AdminDashboard.js.



```
import React from 'react'
import { Container, Row, Col, Button, Tab, Nav, Tabs } from 'react-bootstrap'
import AddScholarship from './AddScholarship'
import ScholarshipTypes from '../user/ScholarshipTypes'
import AdminProfile from './AdminProfile'
import ViewScholarships from './ViewScholarships'
import StudentDetails from './StudentDetails'
import PreviousApplicationDetails from './PreviousApplicationDetails'

const AdminDashboard = () => {
  return (
    <>
      <Tabs
        defaultActiveKey="scholarships"
        id="justify-tab-example"
        className="mb-3"
        justify
        style={{ height: '50px' }}
      >
        <Tab eventKey="scholarships" title="View Scholarships">
          <ViewScholarships />
        </Tab>
        <Tab eventKey="scholarship-form" title="Create Scholarships">
          <AddScholarship />
        </Tab>
        <Tab eventKey="student-details" title="Student Details">
          <StudentDetails />
        </Tab>
        <Tab eventKey="previous-details" title="Previous Applications">
          <PreviousApplicationDetails />
        </Tab>
        <Tab eventKey="profile" title="Profile">
          <AdminProfile />
        </Tab>
      </Tabs>
    </>
  )
}
export default AdminDashboard
```

Fig 5.2.1

## Admin Login- Figure 5.2.2 illustrates the implementation code for *AdminLogin.js*.



```
import React, { useState, useContext } from "react";
import { Container, Row, Col, Form, Alert, Button } from "react-bootstrap";
import userLogin from "../../images/userLogin.jpg";
import { useNavigate } from "react-router-dom";
import axios from "axios";
import { UserContext } from "../../context/auth";
import AdminRegistrationImage from "../../images/userRegistration.jpg"
const AdminLogin = () => {
  const { loggedIn,
    setLoggedIn,
    user,
    setUser,
    adminLoggedIn,
    setAdminLoggedIn, } = useContext(UserContext);

  const navigate = useNavigate();
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const PostData = async (e) => {
    e.preventDefault();
    try {
      const res = await axios.post(`http://localhost:8080/adminlogin`, {
        email,
        password,
      });
      if(res && res.data.success ){
        alert(res.data.message)
        navigate("/adminDashboard")
        setLoggedIn(false);
        setAdminLoggedIn(true)
        setUser({email: email });
        localStorage.setItem("loggedIn", false);
        localStorage.setItem("adminLoggedIn", true);
        localStorage.setItem("user", JSON.stringify(user));
      }else{
        alert(res.data.message)
      }
    } catch (error) {
      // alert(res.data.message)
      console.log(error);
      alert(`Something went wrong ` );
    }
  };
  return (
    <>
    <Container>
      <Row>
        <Col>
          <img src={AdminRegistrationImage} alt="Admin Login" />
        </Col>
        <Col>
```

```

<Form onSubmit={postData} method="POST">
  <h2 style={{ textAlign: "center" }}> Admin Login</h2>
  <Form.Group controlId="formGroupEmail" >
    <Form.Label>Email address</Form.Label>
    <input
      name="email"
      value={email}
      onChange={(e) => setEmail(e.target.value)}
      type="email"
      placeholder="Enter email"
      className="form-control"
      required
    />
  </Form.Group>
  <Form.Group controlId="formGroupPassword" >
    <Form.Label>Password</Form.Label>
    <input
      name="password"
      value={password}
      onChange={(e) => setPassword(e.target.value)}
      type="password"
      placeholder="Password"
      required
      className="form-control"
    />
  </Form.Group>
  <Button type="submit">LOGIN</Button>
</Form>
</Col>
</Row>
</Container>
</>
);
};

export default AdminLogin;

```

**Fig 5.2.2**

**ViewScholarship-** Figure 5.2.3 illustrates the implementation code for *ViewScholarship.js*.



```

import React, { useState, useEffect } from "react";
import { Container, Card, Col, Button, Modal, Row } from "react-bootstrap";
import { Link, useNavigate } from "react-router-dom";
import axios from "axios";

const ViewScholarships = () => {
  const navigate = useNavigate();
  const [scholarship, setScholarship] = useState([]);

  // Get function
  const getScholarships = async () => {

```

```

try {
  const { data } = await axios.get(
    `http://localhost:8080/get-scholarships`
  );
  setScholarship(data.scholarship);
} catch (error) {
  console.log(error);
  alert("Something went wrong");
}
};

useEffect(() => {
  getScholarships();
}, []);

const [show, setShow] = useState(false);

const handleClose = () => setShow(false);
const handleShow = () => setShow(true);

const handleDelete = async() => {
  try {
    const { data } = await axios.delete(
      `http://localhost:8080/get-scholarships/${scholarship[selected]?._id}`
    );
    alert(data.message);
    navigate("/adminDashboard");
    window.location.reload();
  } catch (error) {
    console.log(error);
    alert("Something went wrong");
  }
}

// const [id, setId] = useState();
const [showModal, setShowModal] = useState(false);
// const [data, setData] = useState(data)
const [selected, setSelected] = useState(null);
const handleButtonClick = (index) => {
  setSelected(index);
  setShowModal(true);
};

// const [scholarshipName , setscholarshipName] = useState("");
return (
  <>
  <Container>
  <br/><br/>
  {/* <h1>Scholarships: </h1> */}
  <Row>
    {scholarship?.map((item, index) => (
      <Col sm={12} md={4} style={{ padding: "20px" }}>
        <Card style={{ padding: "20px" }} key={item._id}>
          <h3>{item.scholarshipName}</h3>
          <br />
          <p>Deadline: {item.deadline}</p>
        {/* <p>Last Date to Update: {item.lastDateUpdate}</p> */}
    

```

```

<p>Amount: {item.amount}</p>
<Button
  key={({item._id})}
  onClick={() => handleButtonClick(index)}
>
  View More
</Button>
</Card>
</Col>
))}
</Row>
<Modal show={showModal} onHide={() => setShowModal(false)} size="xl">
<Modal.Header closeButton>
  <Modal.Title>{scholarship[selected]?.scholarshipName}</Modal.Title>
</Modal.Header>
<Modal.Body>
  <Container>
    <Row>
      <Col sm={3} className="modal-label">
        Deadline:
      </Col>
      <Col sm={9}>{scholarship[selected]?.deadline}</Col>
    </Row>
    <br />
    <Row>
      <Col sm={3} className="modal-label">
        Amount:
      </Col>
      <Col sm={9}>{scholarship[selected]?.amount}</Col>
    </Row>
    <br />
    <Row>
      <Col sm={3} className="modal-label">
        Category:
      </Col>
      <Col sm={9}>{scholarship[selected]?.category}</Col>
    </Row>
    <br />
    <Row>
      <Col sm={3} className="modal-label">
        Eligibility:
      </Col>
      <Col sm={9}>{scholarship[selected]?.eligibility}</Col>
    </Row>
    <br />
    <Row>
      <Col sm={3} className="modal-label">
        Documents Required:
      </Col>
      <Col sm={9}>{scholarship[selected]?.documents}</Col>
    </Row>
    <br />
    <Row>
      <Col sm={3} className="modal-label">
        Note:
      </Col>
      <Col sm={9}>{scholarship[selected]?.description}</Col>
    </Row>
    <br />
  </Container>
</Modal.Body>
</Modal>

```

```

        </Container>
        {/* <p>Deadline: {data[selected]?.deadline}</p>
        <p>Last Date to Update Details: {data[selected]?.lastDateUpdate}</p>
        <p>Amount: {data[selected]?.amount}</p>
        <p>Eligibility: {data[selected]?.eligibility}</p>
        <p>Documents Required: {data[selected]?.DocumentsRequired}</p> */}
    </Modal.Body>
    <Modal.Footer>
        <Link to={`/update-scholarship/${scholarship[selected]!.id}`}>
            <Button variant="primary" >Update</Button>
        </Link>
        <Button variant="danger" onClick={handleShow}>Delete</Button>
    </Modal.Footer>
</Modal>

<Modal
    show={show}
    onHide={handleClose}
    backdrop="static"
    keyboard={false}
    centered
>
    <Modal.Header closeButton>
    <Modal.Title>{scholarship[selected]!.scholarshipName}</Modal.Title>
    </Modal.Header>
    <Modal.Body>
        Are you sure You want to delete this Scholarship?
    </Modal.Body>
    <Modal.Footer>
        <Button variant="secondary" onClick={handleClose}>
            Close
        </Button>
        <Button variant="danger" onClick={handleDelete}>Delete</Button>
    </Modal.Footer>
</Modal>
</Container>
</>
);
};

export default ViewScholarships;

```

**Fig 5.2.3**

**Scholarship Schema-** Fig 5.2.4 illustates the implementation code for *Scholarshipmodel.js*.



```
const mongoose = require("mongoose");

const scholarshipModel = new mongoose.Schema({
  scholarshipName: {
    type: String,
    required: true,
    unique : [true, "Scholarship Name already exists"],
  },
  deadline: {
    type: String,
    required: true,
  },
  amount: {
    type: Number,
    required: true,
  },
  category: {
    type: String,
    required: true,
  },
  eligibility: {
    type: String,
    required: true,
  },
  documents: {
    type: String,
    required: true,
  },
  description: {
    type: String
  },
  timestamp: {
    type: Date,
    default: Date.now
  },
});

const Scholarships = mongoose.model("scholarships", scholarshipModel);

module.exports = Scholarships;
```

**Fig 5.2.4**

## CHAPTER 6

### TESTING AND MAINTENANCE

- 1. Unit Testing** – Verifying individual components (e.g., user login, scholarship search). Unit testing is a method of testing where individual parts or functions of a program are tested on their own to make sure they work correctly. It helps catch bugs early, makes the code more reliable, and makes it easier to update later. Developers often use tools like JUnit or pytest to automate these tests.
- 2. System Testing** – Testing the complete system for expected outputs. System testing is a type of software testing where the entire software system is tested as a whole. It checks if the system meets the required specifications and works properly in all intended settings. This includes verifying end-to-end scenarios, covering both functional and non-functional aspects such as performance and security.
- 3. User Acceptance Testing (UAT)** – This helps in ensuring that the portal meets user requirements. The User Acceptance Testing (UAT) is the last step in software testing where real users check the system to make sure it meets their needs and is ready to be launched. It looks at how the software works in real-life situations and confirms that it performs as expected, helping to ensure customer satisfaction before the software is released.
- 4. Integration Testing** – Testing if modules work together(e.g., user authentication + database). Integration testing checks how data flows and how integrated modules interact to make sure they work well together. It helps find interface problems and ensures different parts of the system communicate smoothly.

# CHAPTER 7

## RESULTS AND DISCUSSIONS

### 7.1 Description of modules with snapshots:

The screenshot shows the 'Create Scholarship Form' page of the ScholarAid application. At the top, there is a navigation bar with links for 'View Scholarships', 'Create Scholarships' (which is the active tab), 'Student Details', 'Previous Applications', and 'Profile'. On the right side of the header is a 'Logout' button. The main form area has a title 'Create Scholarship Form'. It contains several input fields: 'Scholarship Name \*' with a placeholder 'Enter Scholarship Name', 'Deadline \*' with a date input field 'mm/dd/yyyy' and a calendar icon, 'Amount \*' with a text input field, 'Category' with a dropdown menu, 'Eligibility Criteria \*' with a placeholder 'Enter Eligibility Criteria', 'Documents Required' with a placeholder 'Enter Documents Required', and 'Extra Information' with a placeholder 'Add Extra information if any'. At the bottom of the form are two buttons: a blue 'Submit' button on the left and a white 'Cancel' button on the right.

**Fig 7.1.1**

-The above module (Fig 7.1.1) provides a form for publishing scholarship by the publisher.

The screenshot shows the 'View Scholarships' page of the ScholarAid application. At the top, there is a navigation bar with links for 'View Scholarships' (active), 'Create Scholarships', 'Student Details', 'Previous Applications', and 'Profile'. On the right side of the header is a 'Logout' button. Below the navigation bar, there is a grid of four scholarship listings. The first three are full-width boxes, and the fourth is a half-width box. Each listing includes the scholarship name, deadline, amount, and a 'View More' button. The first listing is 'Scholarship 2' (Deadline: November 29, 2024, Amount: 4545465). The second is 'UP Scholarship' (Deadline: December 20, 2024, Amount: 10000). The third is 'Ibtesam' (Deadline: September 24, 2024, Amount: 200000). The fourth listing is 'Inspire' (Deadline: September 26, 2024, Amount: 25000).

Scholarship Name	Deadline	Amount
Scholarship 2	November 29, 2024	4545465
UP Scholarship	December 20, 2024	10000
Ibtesam	September 24, 2024	200000
Inspire	September 26, 2024	25000

**Fig 7.1.2**

-The above module (Fig 7.1.2) displays scholarships published by the sponsor.

The screenshot shows a user profile page for 'Deepanshu'. At the top, there's a navigation bar with links for 'View Scholarships', 'Create Scholarships', 'Student Details' (which is the active tab), 'Previous Applications', and 'Profile'. On the right side of the header is a 'Logout' button. Below the header, a table displays a single row of data:

Student Name	Scholarship Name	Category	Amount	Status
Deepanshu	Ibtisam	international	200000	Accepted

A blue 'View More' button is located at the bottom right of the table.

**Fig 7.1.3**

-The above module (Fig 7.1.3) displays applicants in a specific scholarship scheme.

The screenshot shows a grid of six scholarship categories, each represented by a card:

- Merit Based Scholarship**: Includes a small computer icon, a brief description, and a 'View Scholarships' button.
- Need Based Scholarship**: Includes a small computer icon, a brief description, and a 'View Scholarships' button.
- International Scholarships**: Includes a small double-page icon, a brief description, and a 'View Scholarships' button.
- Minority Scholarship**: Includes a small computer icon, a brief description, and a 'View Scholarships' button.
- Research Scholarship**: Includes a small double-page icon, a brief description, and a 'View Scholarships' button.
- All Scholarships**: Includes a small double-page icon, a brief description, and a 'View Scholarships' button.

**Fig 7.1.4**

-The above module (Fig 7.1.4) categorises various available scholarships on the basis of various parameters



## Statistics

**Fig 7.1.5**

-The above module (Fig 7.1.5) is the homepage of the website.

**Eligibility Criteria**

- To be eligible for a scholarship, applicants must meet all of the necessary requirements specified by the scholarship program.
- Eligibility requirements may include maintaining a certain GPA, demonstrating financial need, or being enrolled in a particular course of study.
- It is important for applicants to carefully review the eligibility criteria for each scholarship opportunity they are interested in.
- Applicants should ensure that they meet all of the necessary requirements before applying.
- Meeting the eligibility criteria is a necessary first step in the application process.
- However, meeting the eligibility criteria does not guarantee that an applicant will receive a scholarship.

**Application Process**

- Research scholarships to identify those that you are eligible for and interested in
- Gather application materials, which may include academic transcripts, test scores, letters of recommendation, personal essays or statements, and financial aid information
- Complete application forms accurately and completely
- Submit applications by the deadline specified by each scholarship program
- Follow up with the scholarship provider to confirm receipt of your application or to provide additional information if requested
- Await the decision, which will be based on the eligibility criteria and strength of your application

**Fig 7.1.6**

-The above module (Fig 7.1.6) describes the details about a specific scholarship when clicked on any.

The screenshot shows the ScholarAid website's homepage. At the top, there is a navigation bar with links for Home, Scholarships, Information, Contact Us, Profile, and Logout. Below the navigation bar, there are two rectangular boxes. The left box is titled "UP Scholarship" and contains the text "Deadline: December 20, 2024" and "Amount: 10000", with a "View More" button at the bottom. The right box is titled "Inspire" and contains the text "Deadline: September 26, 2024" and "Amount: 25000", also with a "View More" button.

**Fig 7.1.7**

-The above module (Fig 7.1.7) shows the scholarship applied on the applicant side.

The screenshot shows the ScholarAid website's "Contact us" page. At the top, there is a navigation bar with links for Home, Scholarships, Information, Contact Us, Profile, and Logout. Below the navigation bar, there is a section titled "Contact us" containing several contact details: an email address (support@website.com), a website URL (www.website.com), a phone number (+001) 123 456 7890, operating hours (9:00 AM - 6:00 PM), and an address (1644 Deer Ridge Drive Rochelle Park, NJ 07662). To the right, there is a form with fields for Name, Email address, and Message, each with an input placeholder. A "Submit" button is located at the bottom right of the form.

**Fig 7.1.8**

-The above module (Fig 7.1.2) provides contact information of Scholar Aid.

## **7.2 Technology Discussions:**

### **HTML:**

#### **Introduction:**

HTML full form is HYPER TEXT MARKUP LANGUAGE. It is the standard language used to create and structure content on the World Wide Web. It is the main building block for all websites and web applications that you use in a web browser. Whether it's a simple blog or a complex e-commerce application, HTML is the framework on which web content is created, stored and viewed.

The "HyperText" portion refers to the linking ability (or hyperlinks) between different web documents so that when accessing content via the web, users are able to seamlessly navigate from one web document to another without allowing the lack of connecting web documents hinder their experience. Markup Language is also important, because HTML is not programming language; it is a language that marks up, or annotates, text data with tags that tell the browser how to display content.

#### **Structure of an HTML Document:**

HTML Code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>My First HTML Page</title>
  </head>
  <body>
    <h1>Welcome!</h1>
    <p>This is a paragraph of text.</p>
  </body>
</html>
```

**Fig 7.2.1**

#### **History and evolution of HTML:**

Tim Berners-Lee originally developed HTML in 1991 to share documents on the web, and provided a consistent format for this readable and accessible process. HTML has gone through various iterations from that time, which progressively improved the language to accommodate even richer, and more dynamic, content.

Key milestones in HTML evolution:

### **HTML 1.0:**

This was the first step in 1993.

Think back to when the internet was just beginning. Text based browsers were popular and the rich visual experience we have today was merely a dream. During this fledgling era, the HyperText Markup Language was first defined as HTML 1.0 in 1993. It was very rudimentary but it established the stage for the world we now take for granted.

### **Core Concepts:**

At its essence, HTML 1.0 was created to structure and link documents together. It offered a means to markup text indicating its meaning and how it should be rendered in a primitive way. The basic components included:

**Basic Text Formatting:** Tags like `<h1>` to `<h6>` for headings, `<p>` for paragraphs, `<b>` for bold text, `<i>` for italic text, and `<br>` for line breaks allowed for a minimal level of text styling and organization. Think of it as the digital equivalent of highlighting and indenting in a plain text document.

### **HyperLinks:(`<a>` tags)**

This was a pivotal component that allowed HTML to be "hypertext." The `a` tag was equipped with an `href` attribute that allowed the creation of a link to another document on the same server or on a different server somewhere on this fledgling network. In other words, the ability to connect various pieces of information was instrumental to the capabilities of the World Wide Web.

### **Lists:**

To structure information sequentially or in an unordered manner, HTML 1.0 provided tags for unordered lists (`<ul>`), ordered lists (`<ol>`), and list items (`<li>`). This allowed for better organization of content beyond simple paragraphs.

### **Image:**

Image is done by `<img>` tag.

The use of images, even in a minor way, added an important visual element to web pages. With the `img` tag and its `src` attribute, browsers could request an image file and display it inline to text.

### **Document Structure (`<html>`, `<head>`, `<title>`, `<body>`):**

These foundational tags provided the basic skeleton for an HTML document.

- `<html>` acted as the root element, encompassing the entire HTML content.
- `<head>` contained meta-information about the document, such as the title.
- `<title>` specified the text that would typically appear in the browser's title bar.
- `<body>` contained the visible content of the web page.

### **Limitations:**

1. Layout Control - There were no support for advanced layout elements like tables or css. Web pages represented linear paths of text and images.
2. Interactivity - There were no forms, and no scripting capabilities, so web pages were static displays of information.
3. Styling Options - The presentational options were very basic and dependent on the browser. There were no standard way of controlling colors, fonts, or spacing, other than primitive text formatting called tags.

### **Significance:**

HTML 1.0 was an extraordinary accomplishment, even though it was a quite rudimentary one. HTML 1.0 provided the original building blocks and the critical idea of hyperlinking

which began the development of the World Wide Web. This demonstrated the capability of standardizing a markup language for saving and sharing information around the world.

## **HTML 2.0:(1995)**

Then, in 1995, two years later, the release of HTML 2.0, which was a vital part of the genesis of the web, came with important features supporting more interaction and structure to the content.

### **Key Innovation:**

Forms (`<form>`, `<input>`, `<textarea>`, `<select>`, `<option>`): This was a game-changer. Forms allowed users to interact with web pages by entering data, selecting options, and submitting information back to a server. This paved the way for early web applications, search engines, and basic data collection. The various form elements included:

- `<form>`: Defined an HTML form for user input.
- `<input>`: Allowed for various types of input fields like text boxes, radio buttons, checkboxes, and submit buttons.
- `<textarea>`: Provided a multi-line text input area.
- `<select>` and `<option>`: Enabled dropdown lists for users to choose from predefined options.

Tables (`<table>`, `<tr>`, `<td>`, `<th>`): The introduction of tables provided a structured way to organize and display tabular data. This was essential for presenting information like statistics, schedules, and comparisons in a clear and organized manner.

- `<table>`: Defined a table.
- `<tr>`: Defined a table row.
- `<td>`: Defined a table data cell.
- `<th>`: Defined a table header cell.

**Image Alignment (align attribute):** The `<img>` tag was enhanced with the `align` attribute, allowing for basic control over how text flowed around images (e.g., aligning an image to the left or right of a paragraph).

**Basic Character Entities:** Support for a limited set of character entities was introduced, allowing for the display of characters that were difficult or impossible to type directly.

**Impact:**

HTML 2.0 transitioned the web from being a set of static documents to a more interactive one. Forms allowed for user interaction while tables offered an important element for organizing interactive information. This version provided a baseline for many interactive web experiences we now take for granted.

## **HTML 3.2:(1997)**

In 1997, HTML 3.2 brought new advancements, most importantly the introduction of scripting capabilities, which would fundamentally change the dynamic nature of web pages.

**Significant Additions:**

**Scripting Support:**

With a `script` tag, client-side scripting languages, particularly JavaScript, opened the door to dynamic web pages. Developers could enhance web pages with dynamic behavior, manipulating the Document Object Model (DOM) actions on behalf of the user. They could create a better experience by having the web page interact with the user without communicating with the server. This was a game-changer in the world of web development.

### **Applet Support (<applet> tag):**

While largely superseded by other technologies today, HTML 3.2 introduced the <applet> tag for embedding Java applets into web pages, enabling richer multimedia and interactive applications within the browser.

### **Improved Layout Control:**

HTML 3.2 has introduced some elements and attributes that offered slightly more control over layout, such as the <div> and <span> tags for grouping content, although styling was still effectively done through presentational attributes.

### **Significance:**

HTML 3.2 marks the beginning of the dynamic web. The addition of script tags, primarily for JavaScript, permitting interactive elements, client-side validation and dynamic page content to present a much more interesting and functioning web page. Admittedly, capabilities for layout were still limited by our current standards, however because we could now group like elements for the first time with it gave us a better option for design context.

### **HTML 4.01: (1999)**

HTML 4.01 was released in 1999, introducing a new philosophy, focusing on separating content (HTML) from presentation (styling). The goals of HTML 4.01 were to create cleaner and more semantically represented markup, as well as to allow the use of Cascading Style Sheets (CSS) to become widespread.

### **Key Principles and Features:**

**Strict vs. Transitional Doctypes:** Document type definitions (DTDs) were introduced in HTML 4.01. The "Strict" DTD encouraged authors to write semantically correct HTML while avoiding any presentational attributes, and the "Transitional" DTD allowed for some legacy presentational attributes, resulting in a smoother transition to cleaner markup.

**Emphasis on Semantic Markup:** New elements were introduced to better define the meaning and structure of content, such as `<abbr>` for abbreviations, `<acronym>` for acronyms (though later superseded), `<cite>` for citation, `<blockquote>` for long quotation, and `<label>` for form control label. The focus is on detailing what the context was, rather than how it should look.

**Frames (`<frame>`, `<frameset>`, `<noframes>`):** While now largely discouraged due to usability and accessibility issues, HTML 4.01 introduced frames, allowing web pages to be divided into multiple independent, scrollable windows, each displaying a separate HTML document.

**Internationalization:** Enhanced support for international character sets and languages was also introduced.

**Accessibility Considerations:** HTML 4.01 had a greater emphasis on accessibility, suggesting the use of attributes like `alt` for imagery, which provided textual alternatives for assistive technologies like screen readers.

**Deprecation of Presentational Elements:** Most of the presentational elements and attributes in earlier versions (such as `<font>`, `<text>`, and the `bgcolor` attribute) were deprecated to encourage developers to styling purposes using CSS instead.

## **Impact:**

HTML 4.01 represented a significant moment in efforts to impose order and maintainability on the web. A focus on semantic markup improved accessibility and made web pages more intelligible to search engine crawlers and other user agents. The

encouragement of separation between content and presentation that HTML 4.01 required was a cornerstone for the emergence of CSS or Cascading Styles Sheets. CSS was an incredible advancement on web design, offering superbly functional and adaptable style options.

## **HTML5: (2014 onwards)**

HTML5 comprises an extensive update to the language, providing improved necessities of modern technology on the web, with an emphasis on more rich multimedia, better semantics, better APIs for web applications, and improved mobile support. While HTML5 is a change for the better, it is not an update that you can simply download. HTML5 is a living standard that continues to evolve as new updates will arrive with features.

### **Key innovations:**

New Semantic Elements: HTML5 introduced a wealth of new semantic elements that more clearly describe the structure and meaning of content. Examples are `<article>`, `<aside>`, `<nav>`, `<header>`, `<footer>`, `<section>`, `<main>`, and `<figure>`. This improves SEO, accessibility, and the overall understandability of the document structure.

Multimedia Support (`<audio>` & `<video>`): Native support for embedding audio and video content without relying on third-party plugins like Flash was a major advancement. The `<audio>` and `<video>` tags provide standardized ways to include multimedia, along with APIs for controlling playback.

Canvas (`<canvas>`): The `<canvas>` element provides a dynamic rendering context for drawing graphics, animations, games, and other visual elements using JavaScript.

Scalable Vector Graphics (SVG) Integration: While it was supported in earlier browsers, then also HTML5 formally integrated SVG, allowing for the integration of vector-based graphics that scale without loss of quality.

Web Storage (localStorage, sessionStorage): HTML5 introduced the structure and mechanisms for storing data directly within the user's browser, allowing for offline capabilities and improved performance for web applications.

Geolocation API: This API allows web applications to read the user's geographical location (allowed with their permission), enabling location-based services.

Web Workers: Web Workers allow JavaScript to run in the background and it is separate from the main browser thread, by improving the performance and responsiveness for complex web applications.

Forms 2.0: Enhanced form features were introduced, including new input types (e.g., email, date, number, range), validation attribute, and improved UI elements.

Drag and Drop API: This API enables interactive drag-and-drop functionality within web pages.

Application Cache (AppCache - now largely deprecated): While facing some security and usability concerns and being largely replaced by Service Workers, AppCache aimed to allow web applications to work offline.

Better Mobile and Device Compatibility: HTML5 was designed with mobile devices in mind, addressing issues like touching input and responsive designing (often used in matters with CSS3).

## **Impact:**

HTML5 has revolutionized the contemporary web. The quality of the web as a feature rich, interactive, and accessible place to consume information has transformed with HTML5 making it possible without proprietary plugins. HTML5 has provided us with semantic markup capability, integrated multimedia support, and a powerful suite of APIs; we were then able to build apps on the web that rival desktop apps. HTML5 also pushed for the public's demand for mobile friendliness in this era of always on access via mobile devices. HTML5 remains a living standard, continuously growing, and allowing the web platform to be lively and adaptable with any advancements in technology.

In conclusion, in its relatively brief history, HTML has come a long way from its inception as a basic markup language to the flexible, powerful foundation of the modern web. It has constantly innovated and evolved, building on each previous version as requirements and capabilities grew; this led us to the rich, interactive digital space we have today. Ultimately, HTML, and web technologies, will continue to adapt, so we can continue to innovate web applications, experiences, and consumption styles in the future.

Some of the HTML Tags are as follows:

<b>Tag</b>	<b>Description</b>
<h1>–<h6>	Headings from largest to smallest
<p>	Paragraph
<a>	Anchor/hyperlink
<img>	Image
<ul>, <ol>, <li>	Lists
<div>	Block container
<span>	Inline container
<table>, <tr>, <td>	Tables

<b>Tag</b>	<b>Description</b>
<form>	Form container
<input>, <textarea>, <button>, <select>	Form controls

### **Attributes in HTML:**

HTML elements can have attributes that provide additional information about that element. These are always placed in the opening tag and usually come in name/value pairs.

Code example:

```
<a href="https://website.com" target="_blank">Visit Website</a>
```

-href: Specifies the URL.

-target="\_blank": Opens the link in a new tab.

### **Common Attributes:**

id: Unique identifier.

class: Used for styling with CSS.

src: Source file (for images, scripts, etc.).

alt: Alternative text for images (important for accessibility).

style: Inline CSS.

title: Tooltip text on hover.

### **Semantic HTML:**

Semantic HTML uses meaningful tags that clearly describe their purpose in the document. This improves:

- Accessibility (for screen readers).
- SEO (Search Engine Optimization).
- Maintainability of the code.

Examples of Semantic Tags:

- <header>: Page or section header
- <footer>: Footer information.
- <article>: Independent content unit.
- <section>: Thematic grouping of content.
- <nav>: Navigation menu.
- <main>: Main content of the page.
- <aside>: Sidebar or supplementary content.

## **HTML forms and Input Elements:**

```
<form action="/submit" method="POST">
  <label for="username">Name:</label>
  <input type="text" id="username" name="username" required />
  <input type="submit" value="Submit" />
</form>
```

## **REACT JS:**

At its heart, React.js is a JavaScript library (not a full-fledged framework) used for building user interfaces, especially for single-page applications (SPAs). It was developed by Facebook and released in 2013, and since then, it has exploded in popularity among frontend developers.

If you've ever used a modern web app where things feel smooth and snappy—like Gmail, Instagram, or Facebook—you've likely interacted with an app built using React (or something similar).

React makes it easy to create reusable UI components, manage dynamic data, and keep everything in sync between your JavaScript logic and what the user sees on the screen.

### **Core Idea of Components:**

For example, a "Button" component can be reused everywhere in your app with different labels or actions:

```
function MyButton(props) {
  return <button>{props.label}</button>;
}
```

Now, It is like:

```
<MyButton label="Click Me" />
<MyButton label="Submit" />
```

### **JSX: Javascript meets HTML**

JSX is a syntax extension that lets you write HTML-like code directly in JavaScript. It looks weird at first, but it helps developers visually understand what the component will render.

### **Unidirectional Data Flow:**

React uses a unidirectional (one-way) data flow. This means that data flows from parent components to child components through props. This makes it easier to reason about your application and debug problems.

### **React Hooks**

In modern React, **hooks** are a way to use state and lifecycle methods inside function components (previously only available in class components). Popular hooks include:

- `useState()`: For state variables.
- `useEffect()`: For side effects like fetching data.
- `useContext()`: For sharing global data.

### **React Ecosystem:**

React's core library focuses just on UI, but its ecosystem is rich and extensive:

React Router: Handles navigation between pages.

Redux / Context API: For managing global state.

Styled Components or Tailwind: For styling.

Next.js: A powerful React framework for server-side rendering and routing.

Developers use React JS because it provides:

- Component-based structure
- Reusable UI
- Virtual DOM
- Strong community and job market

### **NODE JS:**

Node.js is a runtime environment that lets you run JavaScript code outside the browser, typically on the server-side. It was built on Chrome's V8 JavaScript engine and released in 2009 by Ryan Dahl.

Before Node, JavaScript was mainly used just for frontend tasks. Node.js changed that by making JavaScript a full-stack language—you can now build both the frontend and backend with the same language.

Node JS is:

- Single-threaded
- Event-driven
- Non-blocking

Use cases of NODE JS are:

- Building REST APIs (using Express.js)
- Real-time apps like chat or collaboration tools
- Microservices architecture
- Command-line tools
- Backend for mobile or web apps

In real-time projects, Node JS works good for:

- APIs that serve React apps
- Authentication and authorization
- Database connections (MongoDB, MySQL, PostgreSQL)
- WebSockets for real-time chat

## **MONGO DB:**

MongoDB has emerged as a top NoSQL database, providing an alternative to the traditional relational database management systems (RDBMS) with its flexible data model, high availability, and performance capabilities. This essay will look at the main

features of MongoDB, its benefits, typical use cases, and its place in the modern data management landscape.

The underlying structure of MongoDB is that of a document database. While relational database stores data in tables with respect to schemas, but, MongoDB stores data in documents, which is allowing for JSON-like data structures. These documents exist within collections, which are equivalent to tables in a relational database. The ability to escape the restrictions of the RDBMS model, which allows for greater flexibility in working with variable data in applications represents one of the key advantages.

Scalability is another important advantage of MongoDB. It was designed to scale horizontally, across many servers using a strategy called sharding. The sharding is a method of distributing data across many devices, let MongoDB to break it up and use large datasets or service a large volume of requests. The distributed nature of the system also provides fault tolerance. If a server goes down it doesn't knock out the entire system. MongoDB also supports replication, taking it a step further by creating multiple copies of the data on different servers.

The final stepping element of MongoDB that helps make it a good choice for growing companies is its rich query language that supports a variety of operations. CRUD (create, read, update, delete) operations as well as complex queries with multiple filter criteria, and data aggregations. Query language MongoDB itself is very denoting it, letting developers retrieve data based on values in fields (values in a range or regular expressions). In addition, there is support for indexing which further enhances the performance of queries since MongoDB (the database) can find specific data using an index instead of having to scan the entire collection.

Beyond any basic query, MongoDB also provides powerful aggregation capabilities. The aggregation pipeline allows developers to perform complex data transformations, such as grouping, filtering, and sorting, in a series of stages. This enables them to derive meaningful insights from their data and generate reports and summaries efficiently.

MongoDB also supports MapReduce, a more flexible but potentially less performant way to perform complex aggregations.

In recent years, MongoDB has become popular in terms of analytics and big data. It is flexible, allowing growth, and can handle unstructured datasets, which allows it to store and process the massive datasets generated by modern applications. While MongoDB is not generally used for complicated analytics queries requiring ACID (Atomicity, Consistency, Isolation, Durability) properties, it can be integrated with other tools, like Hadoop and Spark, in order to perform more complex analytics.

But like any other technology, MongoDB has also some of its limitations. While MongoDB offers good consistency (by default), which means that readers may not always reflect the most recent action, it provides mechanisms for enforcing stronger consistency when needed. Additionally, while MongoDB support transactions, they have evolved over the time, and developers need to be familiar of the specific transaction semantics in their MongoDB version.

In conclusion, MongoDB has revolutionized data management with its flexible schema, scalability, and powerful query capabilities. Its document-oriented model and NoSQL architecture make it a compelling alternative to traditional relational databases for a wide range of modern applications. As data continues to grow in volume, velocity, and variety, MongoDB is well-positioned to play a key role in helping organizations store, manage, and derive insights from their data.

### **Basic Commands in Mongo DB:**

Create/ Switch Database:-

command: use myDatabase

Create Collections:-

db.createCollection("users")

Insert Documents:-

Command:

```
db.users.insertOne({  
  name: "name",  
  age: 30,  
  email: "name@gmail.com"  
})
```

Find Documents:

Command:

```
db.users.find() // All users  
db.users.find({ age: 30 }) // Query by age
```

Update Document:

Command:

```
db.users.updateOne(  
  { name: "name" },  
  { $set: { age: 31 } }  
)
```

Delete Document:-

Command:

```
db.users.deleteOne({ name: "name" })
```

### **Schema Design:**

Embed data: Store related data in the same document.

Reference data: Use manual references (like foreign keys).

Indexing:

Command:

```
db.users.createIndex({ email: 1 })
```

Aggregation:

Command:

```
db.orders.aggregate([
  { $match: { status: "delivered" } },
  { $group: { _id: "$customerId", total: { $sum: "$amount" } } }
])
```

Replication

MongoDB supports **replica sets**, where one primary node receives writes and multiple secondary nodes sync from it. This ensures high availability.

Sharding

MongoDB supports sharding—splitting data across multiple machines. It's useful when you have huge datasets that don't fit on a single server.

## Use Cases:

- Real-time applications (chats, feeds)
- IoT and sensor data
- Product catalogs
- Content management systems
- User profiles and personalization
- MERN stack (MongoDB, Express, React, Node)

## The Power of the MERN Stack in Modern Web Development

The MERN stack has been a very popular and powerful stack for modern web applications. It gives developers a great way to build full-stack applications, utilizing JavaScript at every layer. MERN stands for MongoDB, Express.js, React, and Node.js, which are all significant parts of the application development experience.

At the heart of the MERN stack is MongoDB, a NoSQL database that allows for flexibility when storing data using a JSON-like format. Because MongoDB uses a schemaless structure, developers can easily work with data in a straightforward way without needing to fit it into fixed locations. Express.js is a lightweight web framework that is hosted on Node.js. Because it is web-based, Express.js makes it simpler for developers to run server-side logic and APIs. Express.js will provide the necessary server-side activities, including middleware tasks and routing, while developers focus on the core functionality of the full-stack application.

React is a front-end library used to build an application's user interface. It allows developers to create reusable components that create a dynamic and interactive user experience while efficiently managing the application's state. Node.js is a JavaScript runtime environment that enables developers to run JavaScript code server-side. Using Node.js allows developers to use JavaScript on both the front-end and back-end. In this way, instead of using a separate back-end language, developers can utilize JavaScript from front to back in a unified development workflow.

## **CHAPTER 8**

### **CONCLUSION AND FUTURE SCOPE**

#### **8.1 Conclusion**

So it will ultimately save time for the officer. This is a fully automated system through a web application "Scholarship Management System". Earlier System is maintained fully manually by records of students. Admin can View public documents at anywhere any time. Admin kept all Documents in place and updated the public documents.

In general, the development of a scholarship portal can greatly simplify and increase the management of a scholarship program. In the above synopsis of our project, a portal will be a common platform for students to find scholarships, apply for them and track the progress of their applications. Also there are efficient ways for the administrators to review applications, shortlist candidates, and in the end award scholarships.

#### **8.2 Future Scope**

There is a lot of scope for potential future enhancements of the Scholarship Portal, in order to increase its impact and usability. One major enhancement could be to implement an AI-based recommendation system that will review a student's academic records and scores, income and financial situation, and specific interests to recommend scholarships that they may most likely qualify for. This use of data would enhance personal user experience and success rates for applications.

Another major area of scope is the ability to link the portal to national education databases or identity databases like Digilocker, Aadhaar or universities – this could reduce the verification process to zero preprocessing, a much more efficient way to administer the scholarship processes increasing efficiency and stopping waste and fraud in extraordinary amounts. Furthermore, developing a mobile app will contribute to better accessibility, particularly for potential students in rural areas or with less connectivity.

Another scope for enhancement also include additional partnerships with private companies, NGOs, and other international institutions, which could provide even greater range of scholarship opportunities, especially to study abroad. In addition, there is scope for including chat-bots or live support systems.

There is also potential with the analytics to monitor and study trends relative to submission of applications, reasons for applicants dropping out of the process, and overall success rates of applications relative to performance. Each of these could inform future policy decisions and be considered with each future developments / changes to the content published on the Scholarship Portal.

In terms of incorporating user experience, the Scholarship Portal can continually evolve by seeking meaningful user feedback and by continuously improving the system to be as effective and inclusive as possible for students.

All these ideas foster the potential for the Scholarship Portal to be transformed into a significant instrument for equality in education, development of human capital, and opportunities for students.

### 8.3 Research Paper Status



**Fig 8.1**

**Research Paper accepted**

## 8.4 Patent Status



(Toll Free no.: 1800 1200 106 Mobile: 8588811998, 9911150880)

### Internal Undertaking for Patent/Copyright/Trademark

I/We PAWAN KUMAR PAL (CS)

Main Applicant: KIET GROUP OF INSTITUTIONS DELHI - NCR

Co-applicant / Inventor: PAWAN KUMAR PAL (CS) DEEPANSHU (2125CS1046) HARSHIT MISHRA (2125CS1158) BHITESAM UMAR ANSARI (2125CS1117) JATISH KUMAR (2125CS1109)

I/We have in the course of my study/ employment invented titled, AI Enabled Scholarship Portal :ScholarAid by using the facilities of Institute and I/We are the true and first inventor.

I/We hereby abide by the IPR Policy which was approved by the management and now public to all stakeholders. Also, the intent of research policy of KIET is towards promoting and encouraging Students/Faculties for recognition of their work by promoting their invention through filing patent/copyright/trademark.

I/We are opting the **OPTION - I**

I/We would like to engage with the institute for filing the patent/design/copyright/trademark as per IPR policy. I/We do not have any objection by giving unconditional rights to college (KIET Group of Institutions) to file and register the patent/design/copyright/trademark in their name.

I/We hereby state that we shall be abide by the IPR policy clause no. 8.2, 8.3, 9, 9.1, 9.2, 10, 10.1, a, b, c and 10.2 approved by college management.

I/We do not have any objection by giving unconditional rights to college (KIET Group of Institutions) to file and register the patent/design/copyright/trademark in their name. My/Our claims shall be as per the defined ratio in clause no. 10, 10.1, a, b, c and 10.2 of IPR for sharing revenue if generated through commercialization either by transferring technology fee/ royalty/ one time fee or establishing the venture in future. Both parties shall keep update to each other as per clause and shall abide by the policy.

I/We have given this undertaking at my/our own will and without having any kind of compulsion and pressure by and on behalf of the Institute.

**Signature of the Co-Applicant/Inventor(s)**

**Recommendation of HoD**

HOD.....

Signature.....

Dr. Vibhay Sachan  
(Dean R&D)

Dr. K. Nagarajan  
(Associate Dean Patents)

Approved (Remarks if any) By:

Dr.(Col.) A Garg

**Fig 8.2**

**Patent Filed**

(12) PATENT APPLICATION PUBLICATION  
(19) INDIA  
(22) Date of filing of Application :30/11/2024

(21) Application No.202411094181 A  
(43) Publication Date : 17/01/2025

(54) Title of the invention : WEB PORTAL FOR SCHOLARSHIP DISCOVERY AND MANAGEMENT

(51) International classification	:G06Q0050200000, G06F0021620000, G16H0010200000, G06Q0010100000, G09B0005000000	(71)Name of Applicant : <b>1)KIET Group of Institutions</b> Address of Applicant :Delhi-NCR, Meerut Rd Ghaziabad Uttar Pradesh India 201206 Ghaziabad ----- <b>Name of Applicant : NA</b> <b>Address of Applicant : NA</b>
(86) International Application No	:NA	(72)Name of Inventor : <b>1)Prof. Pawan Kumar Pal</b> Address of Applicant :Computer Science Department, KIET Group of Institutions, Delhi-NCR, Meerut Rd Ghaziabad Uttar Pradesh India 201206 Ghaziabad ----- ----- <b>2)Harshit Mishra</b> Address of Applicant :Computer Science Department, KIET Group of Institutions, Delhi-NCR, Meerut Rd Ghaziabad Uttar Pradesh India 201206 Ghaziabad ----- ----- <b>3)Deepanshu</b> Address of Applicant :Computer Science Department, KIET Group of Institutions, Delhi-NCR, Meerut Rd Ghaziabad Uttar Pradesh India 201206 Ghaziabad ----- ----- <b>4)Ibtessam Umar Ansari</b> Address of Applicant :Computer Science Department, KIET Group of Institutions, Delhi-NCR, Meerut Rd Ghaziabad Uttar Pradesh India 201206 Ghaziabad ----- ----- <b>5)Jatish Kumar</b> Address of Applicant :Computer Science Department, KIET Group of Institutions, Delhi-NCR, Meerut Rd Ghaziabad Uttar Pradesh India 201206 Ghaziabad ----- -----
(87) International Publication No	: NA	
(61) Patent of Addition to Application Number	:NA	
Filing Date	:NA	
(62) Divisional to Application Number	:NA	
Filing Date	:NA	

(57) Abstract :

The present invention relates to a centralized web-based platform designed to streamline the process of discovering, applying for, and managing scholarships for students in India. The platform connects students with available scholarships by matching their profiles, which include personal, academic, and financial details, with scholarships based on predefined eligibility criteria. The system features a student interface for browsing relevant scholarships, a scholarship provider interface for publishing and managing scholarship opportunities, and an administrator interface for overseeing the platform's operations. The matching algorithm ensures that students are presented with scholarships they qualify for, improving the chances of scholarship utilization. The platform also includes features such as real-time notifications, document submission, application tracking, and data security measures to ensure the integrity and privacy of student information. This invention aims to increase accessibility to higher education for students, particularly those from underprivileged backgrounds, by simplifying the scholarship application process and connecting them with funding opportunities from government bodies, NGOs, and private organizations.

No. of Pages : 17 No. of Claims : 10

The Patent Office Journal No. 03/2025 Dated 17/01/2025

6867

**Fig 8.3**  
**Patent Published**

## **REFERENCES**

- 1.HOPE or No-HOPE: Merit-Based College Scholarship Status and Financial Behaviours Among College Students (By Joseph W. Goetz, Yoko Mimura, Miti P. Desai, and Brenda J. Cude)
- 2.The Impact of Scholarships on Students' Academic Performance: A Case of Tertiary Institutions in Enugu State, Nigeria (By Ambrose Nnaemeka Omeje, Solomon Ogbonna Abugu)
- 3.The Academic Performance of Scholarship Students during Medical School (By Glaucia de Oliveira Moreira, Silvia Passeri, Paulo Eduardo Neves Ferreira Velho, Flavio Ferraresi, Simone Appenzeller, Eliana Amaral)
- 4.The China Scholarship Council: An Overview (By Ryan Fedasiuk)
- 5.Do Scholarships Help? Preliminary Results of a Case Study of Students in Scholarship Programmes at Monash University, 1997-2001 (By Donna Aiteken, Eric Skuja, Cathy Schapper)



## PRIMARY SOURCES

1	<a href="http://www.coursehero.com">www.coursehero.com</a> Internet Source	2%
2	<a href="http://ijrpr.com">ijrpr.com</a> Internet Source	1%
3	Submitted to KIET Group of Institutions, Ghaziabad Student Paper	1%
4	<a href="http://vsp1toireaderblogs.indiatimes.com">vsp1toireaderblogs.indiatimes.com</a> Internet Source	1%
5	Submitted to Manipal University Jaipur Online Student Paper	1%
6	<a href="http://www.ijirset.com">www.ijirset.com</a> Internet Source	1%
7	<a href="http://rfh.org.pk">rfh.org.pk</a> Internet Source	<1%
8	Submitted to Melbourne Institute of Business and Technology Student Paper	<1%
9	<a href="http://ouci.dntb.gov.ua">ouci.dntb.gov.ua</a> Internet Source	<1%
10	Submitted to Riga Technical University Student Paper	<1%
11	<a href="http://dev.to">dev.to</a> Internet Source	<1%
	<a href="http://www.afcpe.org">www.afcpe.org</a>	

12	Internet Source	<1 %
13	Submitted to Manipal University Student Paper	<1 %
14	dera.ioe.ac.uk Internet Source	<1 %
15	mgnoida.org Internet Source	<1 %
16	www.staffs.ac.uk Internet Source	<1 %
17	sittechno.org Internet Source	<1 %
18	blogfreedom.ir Internet Source	<1 %
19	Submitted to Gusto University Student Paper	<1 %
20	scouthtml.com Internet Source	<1 %
21	technodocbox.com Internet Source	<1 %
22	web.realinfo.tv Internet Source	<1 %
23	www.prebytes.com Internet Source	<1 %
24	eprints.utar.edu.my Internet Source	<1 %
25	www.avcoe.org Internet Source	<1 %
26	core.ac.uk Internet Source	<1 %

27	elibrary.tucl.edu.np Internet Source	<1 %
28	solveq.io Internet Source	<1 %
29	www.cc.gatech.edu Internet Source	<1 %
30	Sufyan bin Uzayr. "Conquering JavaScript - The Practical Handbook", CRC Press, 2023 Publication	<1 %
31	Submitted to Miva Open University Student Paper	<1 %
32	Mohammad Rezaee, Mostafa. "SaveBirds.App: Democratizing Breeding Bird Survey Data Using Data Science", Bowling Green State University, 2024 Publication	<1 %
33	discol.umk.edu.my Internet Source	<1 %
34	techdim.com Internet Source	<1 %
35	Ronald J. Leach. "Introduction to Software Engineering", CRC Press, 2018 Publication	<1 %
36	vocal.media Internet Source	<1 %
37	www.slideshare.net Internet Source	<1 %
38	cdn.shopify.com Internet Source	<1 %
39	docplayer.biz.tr Internet Source	<1 %

40	growpakistani.pk Internet Source	<1 %
41	webhostinggeeks.com Internet Source	<1 %
42	www.ir.juit.ac.in:8080 Internet Source	<1 %
43	fdocuments.net Internet Source	<1 %
44	qtanalytics.in Internet Source	<1 %
45	webx.ubi.pt Internet Source	<1 %
46	Submitted to American University of Central Asia Student Paper	<1 %
47	Submitted to Coventry University Student Paper	<1 %
48	aws.amazon.com Internet Source	<1 %
49	blog.drake.fm Internet Source	<1 %
50	codingzap.com Internet Source	<1 %
51	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
52	en.wikibooks.org Internet Source	<1 %
53	floqast.com Internet Source	<1 %
	www.mygreatlearning.com	

54	Internet Source	<1 %
55	dokumen.pub Internet Source	<1 %
56	www.gpcet.ac.in Internet Source	<1 %
57	www.lowses.edu.au Internet Source	<1 %
58	Chris Minnick. "Beginning React JS Foundations Building User Interfaces with ReactJS", Wiley, 2022 Publication	<1 %

---

Exclude quotes      Off  
Exclude bibliography      Off

Exclude matches      Off