**SCHOLARSHIP PORTAL**

*(A Project report submitted in the partial fulfillment of the requirements for the award of the Degree of)*

**BACHELOR OF COMPUTER SCIENCE**

**Submitted By**

**A.RODDICK BEVAN**

**(REG NO: 2221128031)**

**&**

**Y.ROSARIYO**

**(REG NO: 2221128032)**

Under The guidance of

**Mrs., R.JENIFER.,MCA.,M.Phil.,** Assistant Professor

****

**DEPARTMENT OF COMPUTER SCIENCE**

**ANANDA COLLEGE**

***(****Accredited with ‘B’ Grade by NAAC****)******(****Affiliated to Alagappa University, Karaikudi****)******(****UGC Recognized under 2(f) and 12(b) Institution****)***

**Devakottai-630303**

**April-2024**

**Mrs., R.JENIFER.,MCA.,M.Phil.,**

Guide

Assistant ProfessorDepartment of Computer Science  
Ananda College  
Devakottai-630303

**CERTIFICATE**

This is to certify that this project entitled “**SCHOLARSHIP PORTAL**” is the bonafide work done by **A.RODDICK BEVAN (2221128031) & Y.ROSARIYO (2221128032)** in partial fulfillment of the award of the degree of Bachelor of Computer Science, under my Guidance.

**Internal Guide Head of the Department**

The **Viva Voce** examination was conducted on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_at Ananda College, Devakottai.

**Internal examiner External examiner**

**A.RODDICK BEVAN Reg.No: 2221128005**

**Y.ROSARIYO Reg.No: 2221128004**Department of Computer Science  
Ananda College  
Devakottai-630303.

**DECLARATION**

I hereby declare that the project certified on “**SHOLARSHIP PORTAL**” submitted in the partial fulfillment of the award of the degree of Bachelors in Computer Science. This is my original work and that has not previously formatted the basis for the award of any degree, diploma or any other similar titles.

**PLACE**: Devakottai **Signature of the candidate**

**DATE:** **(A.RODDICK BEVAN)**

**(Y.ROSARIYO)**

**ACKNOWLEDGEMET**

First and foremost we would like to THANK GOD for the successful completion of this project.

We give our grateful thanks to our parents , family members and friends.

.

We are delightful to thanks **Rev. Dr. S. Sebastian, Secretary** of our college for providing us all the necessary facilities to accomplish this project work successfully.

We express our sincere thanks to **Rev.Dr.S. John Vasantha Kumar, Principal** of our college for his encouragement throughout this project work.

We feel privileged in extending our sense of gratitude to **Mrs.,R.JENIFER.,MCA.,M.Phil., Assistance professor,** Department of Computer Science, Ananda College who served as real guiding and inspiring force to complete this project work.

We also thanks to our system admin **Mr.A.Jeyakumar, DEEE., DCHT., HDCA.,**and Lab Assistant **Mr.P.Hariharan, BA.,B.Ed,** for their encouragement and help.

**A.RODDICK BEVAN**

**S.AATHITHYAN**

**ABSTRACT**

The main goal of the Blood Bank and Donor Management System project is to monitor Blood Bank data, Donor data, Donor list.This is a web-based Blood Donation Bank Management System project developed using **HTML, CSS, JS, PHP and MySQL**.The project is designed to improve the efficiency of blood bank management and provide a transparent and corruption free system for blood donation.

It manages all the Blood Bank, Donors, Donor availability. The project is entirely administrative and therefore access is guaranteed only to the administrator.The project's aim is to develop an application system to minimize the searching time for Blood Bank, Donor, Blood Group management.It monitors all of the Blood Group information, Blood supply and Donor list.

Blood Bank Management Project for track about blood donors and available blood groups created by making use of HTML, CSS, JavaScript, PHP, and MySQL.The application has been developed using MySQL, HTML, CSS, and JavaScript, with the connectivity being achieved through the use of PHP..

This is an attempt to show how a Blood Bank Management system works and also, showcases the data being accessed, retrieved, and manipulated very easily through the help of the user-friendly interface developed.

**TABLE OF CONTENT**

| **SNO** | **INDEX** | **PAGE NO** |
| --- | --- | --- |
| **1** | **INTRODUCTION**   1. Project overview 2. Objectives | 2 |
| **2** | **SYSTEM ANALYSIS**   1. Existing System 2. Proposed System 3. Feasibility Study | 3  4  5 |
| **3** | **SYSTEM SPECIFICATION**   1. Hardware Requirement 2. Software Requirement | 6 |
| **4** | **SOFTWARE DESCRIPTION**   1. Introduction to Frontend 2. Introduction to Backend | 7  15 |
| **5** | **PROJECT DISCRIPTION**   1. Model description 2. Data flow diagram 3. Database design 4. Input design 5. Output design 6. Source code | 18  19  20  22 |
| **6** | **SYSTEM IMPLEMENTAION** | 25 |
| **7** | **SYSTEM TESTING**   1. Unit Testing 2. Integration Testing 3. User Acceptance Testing | 26  29  31 |
| **8** | **CONCLUSION & FUTURE ENHANCEMENT**  8.1 Conclusion  8.2 Future enhancements | 32 |
| **9** | **BIBLIOGRAPHY** | 34 |
| **10** | **APPENDIX**   1. Sample Source Code 2. Screen shots | 35  67 |

**INTRODUCTION**

* 1. PROJECT OVERVIEW

A scholarship is an award of financial aid for student to future their

education. Scholarship are awarded based upon various criteria such as academic merits, financial needs, athletic skill etc. The E-Scholarship deals with different application request for getting scholarship through a wed application and enable student to access scholarship information. E-scholarship are meant to offer financial suppoet to students who are pursuing their studies at college level. The proposed work provides different scholarship information to the student and also they send notification to the student. This project aims to provide an efficient, user-friendly platform that bridges this gap, making scholarships more accessible to students.

The program has been accomplished in Python. The administrator and the user can login using username and password which they can select according to their choice. Only restriction is that multiple person will not be permitted to accept the same username. The project has been developed in HDML as front end and Python, MY SQL as back end which develop to help powerful software.

#### **Objectives**

The primary objectives of the CollegeScho project are:

* To provide a centralized platform where students can browse and apply for a wide range of scholarships.
* To allow scholarship providers to post and manage their scholarship listings easily.
* To simplify the application process by guiding users through each step with clear instructions and requirements.
* To enhance the chances of students finding relevant scholarships by offering personalized recommendations based on their profiles and qualifications.

#### **Scope of the Project**

The scope of the CollegeScho project includes:

* Development of a responsive web application accessible on desktop and mobile devices.
* A user-friendly interface for students to search, filter, and apply for scholarships.
* An administrative portal for scholarship providers to create and manage scholarship opportunities.
* Secure authentication and authorization mechanisms for different user roles (students, providers, and admins).
* Integration with a database for storing user information, scholarship details, and application statuses.
* Notifications and updates for students on the status of their scholarship applications.

#### **Target Audience**

* **Students:** College students seeking financial aid through scholarships.
* **Scholarship Providers:** Educational institutions, non-profits, and organizations offering scholarships.
* **Administrators:** Platform administrators who oversee and manage the overall functionality of the system.

#### **Key Features and Functionalities**

* **User Registration and Authentication:** Secure sign-up and login for students and scholarship providers.
* **Profile Management:** Personalized student profiles showcasing qualifications and achievements.
* **Scholarship Search and Filter:** Advanced search options to find scholarships based on eligibility, category, deadline, and other criteria.
* **Application Process:** Step-by-step application submission, with the ability to upload required documents.
* **Scholarship Management:** Dashboard for providers to post new scholarships, review applications, and manage existing listings.
* **Notification System:** Alerts and notifications for application status updates and new scholarship opportunities.

**SYSTEM ANALYSIS**

**2.1 Existing System**

In order to elicit the requirements of the system ans to identify the elements, input, outputs, subsystems and the procedures, the existing system had to be examine and analyzed in detail. This constitutes the system study. Records, slips, procedures, rules etc were examined thoroughly. The existing system was studied involving a co-operation from the employees who run the system at present. Many students face challenges in finding and applying for scholarship due to scattered information across multiple platforms, lack of personalized recommendations based on eligibility, of personalized recommendations based on eligibility, and Inadequate communication between students and scholarship providers.

Its aim to solve these issues by creating a centralized that connects students with scholarship opportunities, simplifies application processes, and enhances communication between applicants and providers.

2.1.1 DRAWBACKS

To add to this the existing system has to keep a lot of paper documents. Maintaining the paper documents and the related complexities in locating an information is fount to be an overhead especially in this era with possibilities if effective way of information management. The main drawbacks include:

* **Data Privacy and Security:**
* Risk of data breaches exposing personal information.
* Need for strong security to prevent hacking and unauthorized access.
* **Internet Dependency:**
* Requires a stable internet connection to use the platform.
* Not ideal for users in areas with poor connectivity.
* **Not User-Friendly for Non-Tech-Savvy Users:**
* Users who are not comfortable with technology might find it difficult to use.
* **High Development and Maintenance Costs:**
* Expensive to develop and maintain the platform.
* Ongoing costs for hosting, updates, and security patches.
* **Scalability and Performance Issues:**
* May slow down or crash if too many users are online at once.
* Needs good infrastructure to handle growing user numbers.
* **Competition with Existing Platforms:**
* Competing with established scholarship websites is challenging.
* Requires effective marketing to attract and retain users.
* **Risk of Outdated Information:**
* Scholarship details might become outdated if not regularly updated.
* Providers need to actively manage their postings.
* **Dependency on Third-Party Services:**
* Relies on external hosting and cloud services.
* Service interruptions or changes in third-party policies can affect the platform.
* **Bias in Scholarship Recommendations:**
* Recommendation engine might unintentionally favor some scholarships.
* Needs careful monitoring to ensure fairness.
* **Legal and Compliance Challenges:**
* Must comply with data protection laws (e.g., GDPR).
* Risk of legal issues if privacy standards are not met.
* **High Competition and Application Overload:**
* Popular scholarships may receive too many applications.
* High rejection rates could discourage students.
* **User Retention and Engagement:**
* Users may not return after applying for scholarships.
* Needs features to keep users engaged, like notifications and updates.

2.2 PROPOSED SYSTEM

A scholarship portal access to increase for deserving students, reducing the financial burden of education. It enhance visibility and reach for scholarship providers, enabling them to attract more applicants. It helps to increase efficiency and organized management of scholarship listings and applications. The scholarship portal improve user experience through a responsive, and it also navigate the platform easily. Some benefits are given below:

2.2.1 ADVANTAGES

**Centralized Platform for Scholarship Information:**

* Combines various scholarships from different sources into one platform, saving students time and effort in searching.

**Easy Search and Filtering:**

* Users can search for scholarships using filters like category, eligibility, deadline, and location, making it easier to find relevant opportunities.

**Simplified Application Process:**

* Streamlines the application process with online forms and document uploads, reducing paperwork and manual submissions.

**Personalized Scholarship Recommendations:**

* Uses user profile data (e.g., academic background, interests) to recommend scholarships tailored to individual eligibility and preferences.

**Real-Time Notifications and Updates:**

* Notifies users about new scholarships, upcoming deadlines, and application status changes through email and in-app alerts.

**Secure User Authentication and Data Storage:**

* Ensures user data privacy with secure authentication (e.g., JWT tokens) and encrypted data storage.

**Increases Visibility for Scholarship Providers:**

* Helps providers reach a larger pool of potential applicants by listing their scholarships on a widely-used platform.

**Saves Time for Students and Providers:**

* Reduces the time students spend searching for scholarships and simplifies application review for providers.

**Accessible from Anywhere:**

* Being a web-based platform, it is accessible from any device with an internet connection, making it convenient for remote users.

**Reduces Paperwork with Digital Applications:**

* Eliminates the need for physical documents, streamlining the submission and review process.

**User-Friendly Dashboard for Tracking Applications:**

* Offers a dashboard for students to track application status and manage notifications, enhancing user experience.

**Scalable to Accommodate Growing Users:**

* Designed to handle an increasing number of users and data, ensuring long-term usability and growth potential.

**2.3 FEASIBILITY STUDY**

A feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. The objective of feasibility study is acquiring a sense of the scope of the system.

The feasibility study of a project can be ascertained in terms of technical factors, economic factors, or both. A feasibility study is documented with a report showing all the ramification of the project. It is very important to evaluate the feasibility of producing quality software is reduced.

The key factors considered during the feasibility study are:

1. Economic feasibility.
2. Operational feasibility.
3. Technical feasibility.

2.3.1 ECONOMIC FEASIBILITY

Economic abalysis is the most commonly used method for evaluating effectiveness of a system. Cost-benefit analysis is the most im-ortant assessment of economic justification of the project. Cost-benefit analysis delineates the cost for project development and weights them against tangible and intangible benefits of a system. This type of analysis varies with the characteristics of the system to be developed, the relative site of the project, and the expected return on investment. Benefits of a new system are always determinant relative to the existing mode of operation.

Economic feasibility deals about the economic impact faced by the organization to implement the new system. Not only cost of hardware, software etc is considered but also the form of reduced costs. The project, installed ce4rtainly be beneficial since there will be a reduction in manual work, increase in speed of work.

The analysis raises financial and economic question during the preliminary investigation to estimate the following:

* The cost to conduct a full system investigation.
* The cost of hardware and software for the class of application of the project being considered.

To be budget feasible, a proposal for the specific project must pass all these test, otherwise it is not considered as a feasible project. I gathered the details regarding the financial aspects incorporated in the system to make it cost efficient.

2.3.2 OPERTIONAL FEASIBILITY:

Suppose for a moment that technical and economic resources are both judged adequate. The systems analyst must still consider the operational feasibility of the requested project. Operatio0nal feasibility is depended on human resources available for the project and involves projecting whether the system will operate and be used once it is installed. If users are virtually wed to the present system, see no problem with it, and generally are not involved in requesting a new system, resistance to implementing the system will be strong. Chances for it ever becoming operational are low.

2.2.3 TECHNICAL FEASIBILITY

There are a number of technical issues, which are generally raised during the feasibility stage of the investigation. A study of function, performance and constraints gave me the ability to achieve acceptable system. The software required for this system is:

* HTML
* PYTHON
* MY SQL

**3. SYSTEM SPECIFICATION**

3.1 HARDWARE REQUIRMENTS

Processor **:** AMD PRO A4-3350B R4

Processor Speed **:** 2.00GHz Onwards

RAM **:** 4 GB

Hard Disk **:** 500 GB

Monitor **:** LG 23”

Network card **:** Any card can provide a 100mbps speed

3.2 SOFTWARE REQUIREMENTS

Operating system **:** Windows 7**/**8**/**10

Front End **:** PHP Framework

Back End **:** My SQL

UI Design **:** CSS, Bootstrap

Development Tool **:** XAMPP Server

## 4. SOFTWARE DESCRIPTION

4.1 FRONT END

**HTML Overview**

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

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

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997**.**

**Development**

In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners- Lee wrote a memo proposing an Internet-based hypertext system Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes from 1990 he listed "some of the many areas in which hypertext is used" and put an encyclopedia first**.**

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991.It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGMLguid, an in-house Standard Generalized Markup Language (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4**.**

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS**.**

Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the Internet Engineering Task Force (IETF) with the mid-1993 publication of the first proposal for an HTML specification: "Hypertext Markup Language (HTML)" Internet-Draft by Berners-Lee and Dan Connolly, which included an SGML Document Type Definition to define the grammar. The draft expired after six months, but was notable for its acknowledgment of the NCSA Mosaic browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes. Similarly, Dave Raggett's competing Internet-Draft, "HTML+ (Hypertext Markup Format)", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms**.**

FEATURES

HTML contains so many features that it took multiple RFCs (Request for Comments) to describe each one. We will go over some most common features of HTML that are useful in a `developer's life.

1. Simple and user-friendly

Tags are annotations that can be used to write HTML. Tags structure HTML and make it easier for people and browsers to read the content efficiently. They also allow a browser to add CSS (Cascading StyleSheets) to the digital document, resulting in a stunning visual combination.

While HTML includes hundreds of tags, just a few of them are necessary for regular use by a developer. Of course, all tags are important; nevertheless, most are rarely used in normal development.

2. Semantic Structure

This is one of the most awaited features of HTML. HTML5 includes several tags for annotating certain elements for their specialized uses. For example, the <article> tag is used to annotate page content. The <aside> tag represents content that is indirectly related to the major content of the document.

Other notable components are the <header>, <footer>, <div>, the paragraph tag <p>, and the <a> tag, which is most commonly used for page navigation.

3. SEO - Search Engine Optimisation

SEO is by far the most important **USP (Unique Selling Point)** of HTML5. With the introduction of search engines such as Google, Yahoo!, DuckDuckGo, and others, you literally have a wealth of knowledge at your fingertips.

These search engines collect information from the World Wide Web by crawling the internet with computer programs (also known as web crawlers) and mapping keywords with respective documents in which they find it. These web crawlers can assimilate this data only because HTML is highly structured, and you can optimize your web pages for searchability.

You may also employ the semantic structure of HTML for Search Engines by using tags such as <title>, <meta> with description, <header>, etc., to inform the web crawler about the relevant keywords in the article.

4. Local Storage & IndexedDB - Client-Side Data Storage

HTML5 brings significant improvements inclient-sidestorage capabilities. LocalStorage and sessionStorage are important technologies in allowing developers to save data on the client side. These are critical developments in browser storage capabilities because cookies limit data storage size on the client side.

Cookies can be used to store small pieces of information, such as authentication tokens or usernames of the client. Still, when consumers return to the web application, the localStorage API comes into use. Avoid loading some basic information on the client, saving both client and server resources. A developer can use the sessionStorage API to persist information only for that session. This means that the data is erased as soon as the user closes the tab. This functionality is useful for browser-based games or high-security applications with limited session times

IndexedDB extends localStorage's client-side storage capabilities. IndexedDB can store more complex and significant data in the key-value form, while localStorage might potentially store a large amount of data in a key-value form where both are strings.

5. Offline Capabilities (PWA) with Cache API & Service Workers

Consider creating a web application that operates even when the user's internet connection is down. You don't have to imagine it because, with the introduction of Service Workers, IndexedDB, and Cache API, you might make it a reality. Your web application could provide your users with a native-like experience. Many applications, such as Flipkart, already do this, and it's known as a PWA (Progressive Web Application).

Service Workers serve as a way between the user's computer and the internet. They can store files locally, retrieve them when needed using the Cache API, and bring information from the IndexedDB to offer data to the application on demand. When a user's device is not connected to the internet, the Service Worker intercepts these requests and serves them from locally stored data. You can set up your service worker to update these files on a regular or ad-hoc basis.

6. Canvas for Game Development

One of the most significant features of HTML is that you can use [HTML5](https://www.scaler.com/topics/html/html-tags/) to create small games, but if you want to create a good video game, you can use the <canvas> element along with CSS and JavaScript. Canvas allows you to create2D and 3D games. You can make these games as interactive as you'd like.

Platform Independent

HTML runs via a browser, which may be found on almost every device with a basic operating system. If you used mobile phones before smartphones, you'd know that even outdated Nokia phones running SymbianOSthat could open HTML pages.

The <figure> tag is particularly worth mentioning because it has revolutionized browser picture rendering. Captions can also be represented using the <figcaption> tag along with the <figure> tag.

8. Media Support

HTML can display images, videos, and audio. Hence it has good media-playback capabilities.HTML5 introduced <video> and <audio> tags, making this much easier than before. Of course, HTML5 allows you to do more than just play media; you can provide controls, graphics for buttons, and even control playback programmatically.

HTML USES:

[HTML](https://www.scaler.com/topics/html/introduction-to-html/) stands for HyperText Markup Language, and it is used to create web pages and web applications. Let's define Hypertext Markup Language (HTML) and a Web page.

**HyperText:** HyperText is an abbreviation for "Text inside Text." A hypertext is text that contains a hyperlink. You have utilized hypertext when you click on a link that leads you to a new webpage. HyperText is a technique for linking two or more web pages (HTML documents) together.

**Markup language:** A markup language is a computer language used to apply style and formatting principles to text documents. Markup language enhances the interactive and dynamic nature of the text. It can convert text into graphics, tables, links, etc.

**Web Page:** A web page is a document that is typically written in HTML and translated by a web browser. An URL is used to locate a web page. A web page can be of two types, i.e., static or dynamic. Static web pages can be created entirely using HTML.

As a result, HTML is a markup language that is used to generate attractive web pages with the help of styling that appears in a nice format on a web browser. An HTML document is made up of several [HTML tags](https://www.scaler.com/topics/html/html-tags/), each with its own set of content.

**Syntax of the HTML tag:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title> Syntax of the HTML </title>

</head>

<body>

...

</body>

</html>

**Markup**

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references**.** HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>**.** The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags)**.** Another important component is the HTML document type declaration, which triggers standards mode rendering. The following is an example of the classic Hello world program, a common test employed for comparing programming languages, scripting languages and markup languages**.**

This example is made using 9 lines of code:

<!DOCTYPE html>

**<html>**

**<head>**

**<title>**This is a title**</title>**

**</head>**

**<body>**

**<p>**Hello world!**</p>**

**</body>**

**</html>**

**Elements**

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: <p>**.** In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" <p> and "end tag" </p>. The text content of the element, if any, is placed between these tags. Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element. The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the <img> used to embed images, the reference to the image resource.

**CSS Overview**

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

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate css file, and reduce complexity and repetition in the structural content. Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified. Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

**sources**

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.

The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. This process is called cascading. One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet. Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

**Example**

Consider this HTML fragment**:**

<!DOCTYPE html>

**<html>**

**<head>**

<**meta** charset="utf-8">

**<style>**

**#xyz** {color: red; }

**</style>**

**</head>**

**<body>**

<**p** id="xyz" style="color: blue;"> To demonstrate specificity

**</p>**

**</body>**

**</html>**

In the above example, the declaration in the style attribute overrides the one in the <style> element because it has a higher specificity.

**Browser support**

Each web browser uses a layout engine to render web pages, and support for CSS functionality is not consistent between them. Because browsers do not parse CSS perfectly, multiple coding techniques have been developed to target specific browsers with workarounds (commonly known as CSS hacks or CSS filters). Adoption of new functionality in CSS can be hindered by lack of support in major browsers. For example, Internet Explorer was slow to add support for many CSS 3 features, which slowed adoption of those features and damaged the browser's reputation among developers.In order to ensure a consistent experience for their users, web developers often test their sites across multiple operating systems, browsers, and browser versions, increasing development time and complexity. Tools such as Browser Stack have been built to reduce the complexity of maintaining these environments.

In addition to these testing tools, many sites maintain lists of browser support for specific CSS properties, including Can I Use and the Mozilla Developer Network. Additionally, the CSS 3 defines feature queries, which provide an @supports directive that will allow developers to target browsers with support for certain functionality directly within their CSS. CSS that is not supported by older browsers can also sometimes be patched in using Javascript polyfills, which are pieces of Javascript code designed to make browsers behave consistently. These workarounds-and the need to support fallback functionality- can add complexity to development projects, and consequently, companies frequently define a list of browser versions that they will and will not support.

**Vertical Control limitations**

Though horizontal placement of elements was always generally easy to control, vertical placement was frequently unintuitive, convoluted, or outright impossible. Simple tasks, such as centering an element vertically or placing a footer no higher than bottom of the viewport required either complicated and unintuitive style rules, or simple but widely unsupported rules. The Flexible Box Module improved the situation considerably and vertical control is much more straightforward and supported in all of the modern browsers. Older browsers still have those issues, but most of those (mainly Internet Explorer 9 and below) are no longer supported by their vendors.

**Absence of expressions**

There was no standard ability to specify property values as simple expressions (such as margin-left: 10% 3em + 4px;). This would be useful in a variety of cases, such as calculating the size of columns subject to a constraint on the sum of all columns. Internet Explorer versions 5 to 7 support a proprietary expression() statement, with similar functionality. This proprietary expression() statement is no longer supported from Internet Explorer 8 onwards, except in compatibility modes. This decision was taken for "standards compliance, browser performance, and security reasons".However, a candidate recommendation with a calc() value to address this limitation has been published by the CSS WG and has since been supported in all of the modern browsers.

**JAVASCRIPT:**

JavaScript is a programming language primarily used to add interactivity and dynamic behavior to web pages, allowing for features like animated elements, user input responses, and dynamic content updates; it was developed by Brendan Eich in just ten days in 1995, instantly called “Mocha” and later “Live Script”, before setting on JavaScript; it was first implemented in the Netscape Navigator browser, quickly gaining popularity due to its ease of use and role as the only client-side scripting language at the time; over the years, JavaScript has evolved significantly through standardization under the ECMA-262 standard, enabling consistent implementation across different browsers and expanding its use beyond web pages to include

Server-side applications and other environments like Node.js.

DEVELOPMENT

**Origin:** Created by Brendan Eich at NetScape communication in 1995.

**Initial Name:** “Mocha” and “Live Script” before setting on “JavaScript”.

**Primary purpose**: To add interactive elements to static HTML pages on the client-side.

**Standardization:** ECMA international standardized JavaScript as “ECMAScript” to ensure compatibilityacross browser.

**Mordern JavaScript features**: Continuous evolution with new features like arrow function, classes, modules, async\await, and more significantly enhancing development capabilities.

**4.2 INTRODUCTION TO BACKEND**:

**MySQL**

MySQL is an open-source relational database management system (RDBMS).Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64, A port of MySQL to OpenVMS also exists.

**History**

MySQL was created by a Swedish company, MySQL AB, founded by David Axmark, Allan Larsson and Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from mSQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use MySQL instead of the (proprietarily licensed) mSQL antecedent.

The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, beginning from 28 June 2000 (which in 2009 has been extended with a FLOSS License Exception) or to use a proprietary license.Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organisations exist to provide support and services, including MariaDB and Percona.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded sql database server"

**Features**

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

* A broad subset of ANSI SQL 99, as well as extension
* Cross-platform support
* Stored procedures, using a procedural language that closely adheres to SQL/PSM
* Triggers
* Cursors
* Updatable views
* Online DDL when using the InnoDB Storage Engine.
* Information schema
* Performance Schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.
* A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
* X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine
* Transactions with save points when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
* ACID compliance when using InnoDB and NDB Cluster Storage Engines
* SSL support
* Query caching
* Sub-SELECTs (i.e. nested SELECTs)
* Built-in replication support (i.e., master-master replication and master-slave replication) with one master per slave, many slaves per master.Multi-master replication is provided in MySQL Cluster, and multi-master support can be added to unclustered configurations using Galera Cluster
* Full-text indexing and searching
* Embedded database library
* Unicode support
* Partitioned tables with pruning of partitions in optimizer
* Shared-nothing clustering through MySQL Cluster
* Multiple storage engines, allowing one to choose the one that is most effective for each table in the application
* Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB Cluste.

**PYTHON**

## Introduction to Python

Python is basically a high-level[programming language](https://www.toppr.com/guides/computer-amplitude-and-knowledge/computer-applications/programming-languages/#:~:text=Other%20examples%20include%20LISP%2C%20ALGOL,examples%20of%20some%20programming%20languages.). It is a dynamic and free open-source language in nature. Moreover, it uses an interpreter for converting the source code into machine code. Furthermore, it supports both object-oriented programming as well as procedure-oriented programming. It is such a language that is highly readable and uses English keywords.

Python is a widely used general-purpose, high-level programming language. It was initially designed by **Guido van Rossum**in **1991**and developed by Python Software Foundation. It was mainly developed to emphasize code readability, and its syntax allows programmers to express concepts in fewer lines of code.

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application-based work in December of 1989 at Centrum Wiskunde & Informatica (CWI) which is situated in the Netherlands. It was started as a hobby project because he was looking for an interesting project to keep him occupied during Christmas.

The programming language in which [Python](https://www.geeksforgeeks.org/python-programming-language)is said to have succeeded is ABC Programming Language, which had interfacing with the [Amoeba Operating System](https://www.geeksforgeeks.org/what-is-amoeba-definition-structure-classification-nutrition)and had the feature of exception handling. He had already helped create ABC earlier in his career and had seen some issues with ABC but liked most of the features. After that what he did was very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and created a good scripting language that had removed all the flaws.

It is derived from many languages like C, C++, Algol-68, ABC, Unix shell, etc. Moreover, it is copyrighted.

## Features

The features are as follows:

### Easy to code and maintain

Many people call it developer-friendly because it is quite easy to learn. Furthermore, we can learn to write its code very easily in a few hours or days. Moreover, this is possible because it has only a few keywords, a simple structure, and a clearly defined syntax.

### Open source and free

It is an open-source language which means that anyone can download it, use it, and share it. Moreover, it is free of cost.

### Object-oriented

It supports object-oriented programming language features. For example, the concept of object and classes, encapsulation, inheritance, etc.

### Supports GUI programming

Python has support for creating various GUI applications. Furthermore, these applications can work in many system software and libraries. Besides, it has modules like PyQt5, PyQt4, wxPython, or Tk through which we can make graphical user interfaces. Among them, PyQt5 is the most popular for making graphical apps.

### Extensible in nature

It is extensible in nature which means that we can use python code in other languages. For example C, C++ also can compile that code in C or C++.

### Portable and Platform independent

It is portable in nature. Moreover, we can run the python code in any environment like windows, Linux, mac, etc without changing it. Besides, if we write the code in one OS and run it on the other it is totally valid.

### Integrated Language

We can easily integrate it with other languages such as C, C++, etc. Hence, it is an integrated language.

### Interpreted Language

It uses an interpreter for converting the source code into machine code. This means that we execute the python code line by line. Hence, it becomes easy to debug the error and solve it.

### Huge Standard Library

There are a very large number of libraries in python. These libraries contain predefined modules and functions for certain tasks. Hence, it becomes easy fr the programmer to develop the code since he does not have to do all the things by himself. Moreover, the library is portable and cross-platform compatible.

### Dynamically Typed

This means that we do not have to define the type of the variable. The interpreter decides it itself at the run time. Hence, the name dynamically typed.

### Scalable

It provides good support for large programs.

Some additional features besides these are as follows:

* It supports both object-oriented and procedure-oriented programming structure.
* We can use it as a scripting language for large applications.
* It contains automatic garbage collection.
* It performs dynamic type checking.
* Case sensitive in nature.
* Useful for developing web applications.
* it uses indentation for program structuring. Unlike other languages that use braces for the distinction of blocks.

## Execution Modes in python

As we know that python uses an interpreter for the execution of source code. now, there are two ways in which we can use the interpreter. They are as follows:

* Interactive Mode
* Script Mode

The interactive mode allows us to execute a single statement instantly. Whereas, in script mode we can write multiple lines of code and then execute it.

### Interactive Mode

In this mode, we can execute a single statement at a time. Moreover, to use the interactive mode, we have to write the statement in front of ‘>>>’ and press enter. This results in the output of that particular statement immediately. This mode is easy and convenient to use to see the instant output. But, at the same time, we cannot save the whole code and have to write it again and again to execute it.

### Script Mode

In this mode we have to write the whole source code and save it as a Python source code file. Furthermore, we can execute this file using the interpreter. Moreover, we save the python source code file with the extension ‘.py’.

**5. PROJECT DESCRIPTION**

**5.1 MODULE DESCRIPTION**

* **User Authentication**
* **Database Management**
* **User Interface (UI)**
* **Business Logic**
* **User Authentication**: Handles user login, registration, and password management.
* **Database Management**: Manages data storage, retrieval, and updates.
* **User Interface (UI)**: Displays information and collects user input.
* **Business Logic**: Processes input data and applies rules or calculations.

**5.2 ARCHITECTURE DESIGN**

**COLLEGE SCHOLARSHIP PORTAL**

USER APPLICATION ADMIN

MODUEL MODULE MODULE

Registration Form Submission uplode

Review

Login Document

Approval

5.3 DATA FLOW DIAGRAMES

The data flow diagrams(DFD) is one of the most important tools used by system analysis. Data flow diagrams are made up of a number of symbols, which represent system components. Most data floiw modelling methods use four kind of symbols. These symbols are used to represent four kind of system components. Possesses data stores, data flows and external entities are the components. Circles in DFD represent a process. Data flow represented by a thin line in DFD aand square or rectangle represents external entities.

Unlike detailed flow chart, data flow diagrame do not supply detailed descripotion of the modules but graphically describes a systems data and how the data interact with the system. In the normal convention a DFD has four major sympbols.

Symbols used in DFD are:

Square, this defines sources or destination of data

Arrow, which shows data flow

Circle, which represent a process that transforms incoming data into

outgoing flow

Open rectangle, which shows a data store

DFD

E-SCHOLARSHIP

DFD LEVEL 0

request response

E-

ADMIN scholarship ADMIN

5.5 DATABASE DESIGN

DATABASE NAME: College scholarship

Primary key\*

Foreign key\*\*

Registration:

| FIELD NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
| --- | --- | --- | --- |
| \*Id | Int(4) | PRIMARY KEY | Registration ID |
| Name | Varchar(100) | NOT NULL | Name |
| Address | Varchar(100) | NOT NULL | Address |
| Phone | Varchar(10) | NOT NULL | Phone |
| Email | Varchar(50) | NOT NULL | Email |
| District | Varchar(10) | NOT NULL | District |
| Password | Varchar(50) | NOT NULL | Password |
| Status | Varchar(50) | NOT NULL | Status |

LOGIN

| FIELD NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
| --- | --- | --- | --- |
| \*Id | Int(4) | PRIMARY KEY | Login ID |
| Username | Varchar(50) | NOT NULL | Username |
| Password | Varchar(50) | FOREIGN KEY | Password |
| Usertype | Text | NOT NULL | Usertype |
| Status | Text | NOT NULL | Status |

CATEGORY

| FIELD NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
| --- | --- | --- | --- |
| \*Id | Int(4) | PRIMARY KEY | Login ID |
| Category | Varchar(100) | NOT NULL | Category |

STUDENT