**CHAPTER 1**

**INTRODUCTION**

* 1. **PROJECT OVERVIEW**

Student space project is a web portal where students and teachers can login to that website, each student and teachers have a different dashboard. Students have the option to see their Attendance details and their personal information. And teachers have the options to add the student, view student Details, take attendance for the particular subjects. This project helps to make the attendance taking procedure for teachers. And it helps the student to track their attendance.

* 1. **PURPOSE OF THE PROJECT**

Student space is the project to take and manage the student attendance and some personal details.

This will help teachers to manage the student attendance details.

Teachers can easily track the student attendance

**CHAPTER 2**

**REQUIREMENT ANALYSIS**

**2.1 EXISTING AND PROPOSED SYSTEM**

**2.1.1 EXISTING SYSTEM**

According to a survey most of the colleges use the paper or book to take and maintain the record of the attendance.it is sometimes very hard to compute the attendance and maintain the attendance details in the college or the particular department, here Students can’t track their attendance details.

**2.1.2 PROPOSED SYSTEM**

In the proposed system, it mainly deals with the web based Attendance system.it helps to take attendance off the students of the particular department.in this also teachers easily track the attendance of the students, who comes under the particular department, also students can check about their attendance details and personal details.

**2.3 TOOLS AND TECHNOLOGIES USED**

**2.3.1 MY SQL**

**MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold vast amounts of information in a corporate network. In particular, a relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure. An RDBMS is simply the set of software tools used to actually implement, manage, and query such a database.

**2.3.2 PHP**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, and even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

**2.3.3 HTML**

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

**HyperText:** Hypertext simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hypertext is a way to link two or more web pages (HTML documents) with each other.

**Markup language:** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

**Web Page:** A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. With the help of HTML only, we can create static web pages.

**2.3.4 CSS**

Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

**2.4.1 PROGRAMMING LANGUAGE TO BE USED:**

➢Front-end language: HTML/CSS

➢Back-end languages: PHP

➢Database: MySQL

**2.4.2 HARDWARE REQUIREMENTS:**

➢Minimum of 512 MB RAM

➢Minimum of 20 GB hard drive space

➢Compatible operating systems:

• X86 Linux operating systems

• 64-bit Windows operating systems (7-latest)

➢Intel x86 or compatible processor

**2.4.3 SOFTWARE REQUIREMENTS:**

➢**IDE:** Visual Studio Code

➢**Browser:** Google Chrome, Mozilla Firefox

➢**Software to manage database:** MySQL

**CHAPTER 3**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**3.1 INTRODUCTION**

SRS totally characterizes the entire venture and the product and how it will be relied upon to perform. A Software Requirement Specification is a whole elaboration of the real constancy and climate for programming under measure. To characterize the customer necessities, designers should have flawless and clear data about items which must be built. It is the fundamental stage for the application progress activity. SRS is the whole meaning of the presentation of association should be set up.

Attributes of SRS:

• Understanding the necessities

• Verification is finished

• Unambiguous

• Consistent is kept up

• Fully Complete

**3.2 FUNCTIONAL REQUIREMENTS**

* **Admin**

Admin who manages the main activity of the web page, who can take the attendance of the student, also add the student details, view the student details. View all the students added by the admin, the admin can check the attendance status of the student, add the subject.

* **Students**

Students are the normal users where students have to login to see their particular details, like personal details ,attendance details etc…,Students also have the option to login to the student dashboard, with the provided username and password.

**3.3 NONFUNCTIONAL REQUIREMENTS**

The applicability of a structure or one of its components is described by a practical needs record. It also depends on the type of programming, target audience, and platform on which the product is used. While useful client requirements may be high-level declarations of what the framework should accomplish, useful framework requirements should also clearly depict what the structure should do.

**Performance**

This page can provide the best response time to the users, because it is used by the teachers and limited students so because of the less traffic the response time became better.

**Usability**

This can be used very easily by the users, because it uses a simple interface which everyone can easily understand without any confusion, it provides the user friendly environment to the users.

**Scalability**

This also supports the future updates to the system, if the hardware is updated, it supports the new specification.

**Portability**

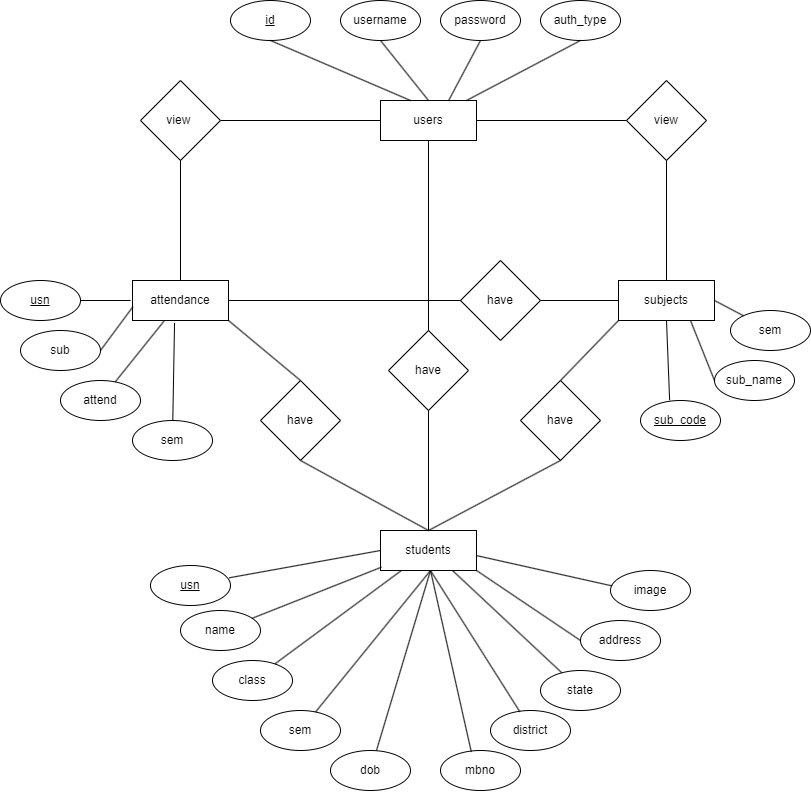
It is also portable, because it uses a simple programming language, it can work on any system.

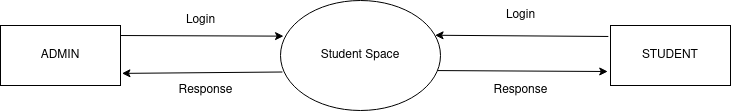
It only needs some software to run this project.

**CHAPTER 5**

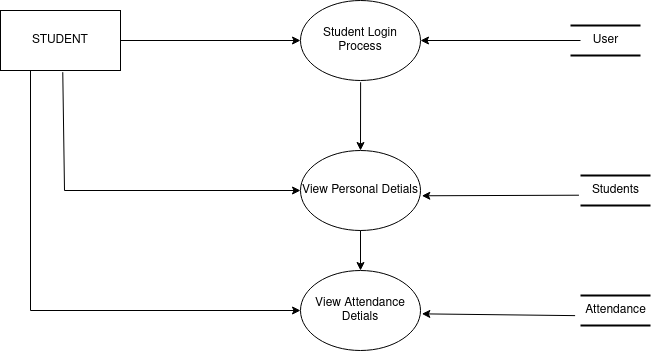
**DETAILED DESIGN**

**5.1 ER-Diagram**



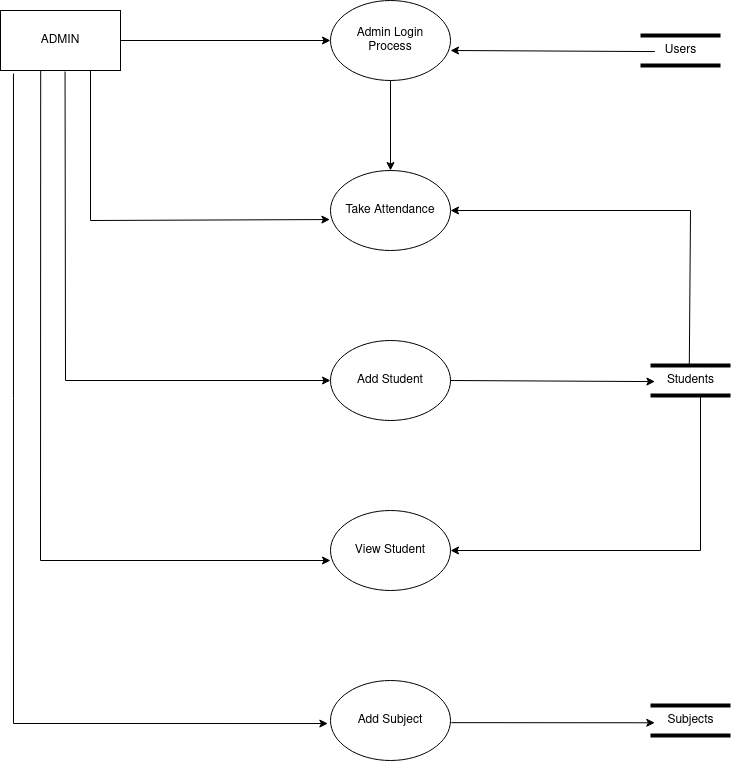
**5.2 DFD Diagram: Level 0**

This is the level 0 DFD diagram, which shows the students and admin logging in using the student space to retrieve the some of the details

**5.3 DFD Diagram: Level 1.1**

This is the level 1 diagram, in that level 1.1 diagram which shows the flows of the students in the student space.in here the student using the credentials to login to the page and viewing the personal information and attendance details.

**5.4 DFD Diagram: Level 1.2**

This is the level 1 diagram, in that level 1.2 diagram which shows the flow of the admin.here admin login and after retrieve the related details in the admin dashboard.

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 Login page**

* The login page is used by both Admin And students
* According to the credentials it redirect them to the respective dashboard
* It contain two input text for entering the username and password
* It also contain a button in the page, The button is pressed by the users
* If the user doesn't fill the fields it displays the error message as please enter the username and password, these are placed inside the card.
* These are placed inside the card, the cards are used for better presentation, so that it becomes highlighted and it becomes easy for the users to read the content which is placed inside the cards.
* In the top of the card the logo is placed
* The card Is placed middle of the page
* This page is completely responsive
* If the user entered the wrong username and password then the page displays the invalid credentials entered.
* This page is built with the html, bootstrap and css.
* The login activity is performed with the sessions.
* When the user logs the sessions are created, and when the user logs out the sessions are destroyed.
* When the sessions are deleted then the users cannot go back to the respective dashboard without re-login.

**6.2 Admin**

**6.2.1 Dashboard**

* After the admin logs in to the login page it redirects the admin to the admin dashboard.
* The admin dashboard contains a sidebar with many option
* And the admin page contains some cards which contain the different details of the student and others.
* The dashboard have the heading ‘dashboard’ in the top of the page
* The side bar can be expanded and compressed. By default it is in the compressed stage. The admin can expand the sidebar by pressing the burger icon.
* The side navbar contains the different icons, which is to represent the different options for the admin.
* The first option is the take attendance, with the icon, while pressing the icon it redirect the admin to the attendance page
* Second one is the add student with an icon, when the admin presses that, it redirects to adding the new student page, where the admin can add the new student.
* Third is the view student details with icon, it redirect the admin to the student details page where all the student details is displayed

**6.2.2 Take Attendance**

* Attendance taking page contains the a table, which contains the Sno, USN, Name, and a radio buttons for marking the present and absent
* it contains the heading in the page as ‘Student Attendance’
* in this page it also displays the current time and date
* it has a dropdown to select the subject for the attendance
* Each student contains two radio buttons for marking the present and absent.
* According to the present and absence the color of the radio button changes.
* By default the present button is checked.
* The present button color is with green, And the absent button color is with red while selected.
* This page has a submit button, after marking the attendance the admin can press the button then the attendance will be recorded to the database.

**6.2.3 Add Student**

* The add student page contains the a card
* Inside the card the input fields are placed
* The card have the heading as ‘enter the student details’
* The form is used to pass the data
* The form contains the full name of the student
* Usn of the student, Class of the student to select, Current semester of the student to select, Date of birth of the student to pick from the calendar, District of the student to pick, State of the student to pick, Mobile number of the student, Address of the student, Image/photo of the student
* These all fields are required field
* The admin can’t leave the fields as empty
* If the admin entered all the fields then it will allow to insert the details to the database
* If the admin enters the already existing student usn it displays the warning message that the ‘student already exists’, Or it will display the message as ‘new student

**6.2.4 View student**

* After adding the students the admin can view the students in the view student page
* The name,usn,image of the student will be displayed in that page
* The details are displayed inside the card, It has a button called view
* While pressing that button it redirects the admin to view the particular information of the student.
* This page is fully responsive
* It changes the alignment of the cards depending on the display size.

**6.2.5 Particular Student information**

* while the admin clicks the view button in the view student page, it redirect the admin to this page
* it contains Some personal details of the student
* the personal details are displayed is the data entered while adding the student
* the student details are displayed in the card along with image of the student
* in this page it also have the attendance details of the student
* it contains a progress bar to indicate the percentage of the attendance, progress bar is used to display the percentage of the attendance in the more graphical way
* it displays the suitable message depending on the percentage of the attendance, like in attendance shortage, clear full attendance.

**6.2.6 Add Subject**

* In this page they can add the new subjects.
* In this page admin can enter the name of the subject, subject code, for which semester the subject belongs to.
* The page has an add button, when the admin presses the add button it adds the subject details to the database.
* This page contains two text input boxes and a dropdown for entering the name, subject code, semester of the subject.
* These elements are placed inside a card.
* The card places the items in the proper position, it highlights the page.
* And it provides a good user interface when using the card.

**6.3 Student**

**6.3.1 View Personal Details**

* The students can log through the login page.
* It redirects them to the student dashboard.
* The students can view their personal details on the page.
* The student details are placed inside the card, which highlights the details part, which contains the image of the student ,name of the student,usn,class,sem, of birth, phone number,District,state Address of the students.
* The student can also view the attendance details on the page.
* The attendance details are also placed inside the card.
* The card contains the name of the subject, attendance status, like attendance shortage, attendance are clear, full attendance.
* It also contains the attendance percentage, and a progress bar which increases according to the attendance percentage.
* The page is divided into 2 parts, one part is viewing the personal details of the students, the second part is the attendance details part, where the details of the student attendance are displayed according to individual students.
* The student dashboard has a logout button which logouts the students from the dashboard.
* The login and logout are implemented using the sessions.
* When the student logs the sessions are created and when the student logouts the sessions are deleted or destroyed.
* If the session is destroyed then the student cannot view the dashboard without re-login.
* This page is fully responsive, when the device width changes it adjusts the contents according to that.

**.**

**CHAPTER 7**

**SOFTWARE TESTING**

**7.1 INTRODUCTION**

Software Testing is done to ensure that the completed programming bundle performs as expected based on the assumptions defined by the requirements/details. The overall purpose isn't to find every product bug that exists, but to uncover conditions that could have a negative impact on the client, ease of use, or viability.

**7.1.2 AIM OF TESTING**

• Identifying surrenders that the developer may make while developing the product.

• To prevent absconds.

• To ensure that the final product satisfies the needs of both the business and the client.

• To ensure that it meets the SRS (System Requirement Specification) requirements.

• To gain the trust of customers by providing a high-quality product.

**7.1.3 Testing Types**

* White box Testing
* Black box Testing

**7.1.4 Levels Of Testing**

* **Unit Testing**

Unit testing is a type of component testing where individual units are tested. Its aim is to ensure that every element of the product functions properly.

* **Integration Testing**

Integration Testing: Here we combine the individual units and they are tested by combining the individual units into groups. The goal of this level of testing is to find weaknesses in the coordination of coordinated units.

* **System Testing**

Framework Testing is a whole, coordinated framework/programming is tested at this level of product testing. The goal is to check whether the framework complies with the predefined requirements.

* **User Acceptance Testing**
* All useful prerequisites are fulfilled.
* All execution prerequisites are accomplished.
* Other necessities like mobility, similarity, mistake recuperation and so on are fulfilled.
* Acceptance rules determined by the client are met.

**7.2 TEST CASES**

**Home Page/Login page**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On Load of Page | It displays login form for admin and students. | Successful |
| 2 | On click of Login Button. | It navigates to the next page if it satisfies login criteria. | Successful |
| 3 | If any mandatory fields left empty | It displays prompt message for user | Successful |
| 4 | If any fields with wrong credentials | It displays prompt message for user | Successful |

**Students view**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On Load of Page | It displays the personal details and attendance of the students. | Successful |
| 2 | On click of Logout Button. | It navigates to login page | Successful |

**Admin Dashboard**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On load of page | It displays an admin dashboard with side navbar and some information. | Successful |
| 2 | On click of Attendance icon | It navigates to student attendance taking page | Successful |
| 3 | On click on Add Student icon | It will navigate to add a student page. | successful |
| 4 | On click on View Student icon | It will navigate to the View student page. | successful |
| 5 | On click on Add Subject icon | It will navigate to the add Subject page. | successful |

**Attendance page**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On load of page | It displays a list of student information in the table. | Successful |
| 2 | On click of present or absent radio button | It selects or mark the student attendance status | Successful |
| 3 | On click on Submit button | It will take or insert the attendance status to the database and navigate to add a dashboard. | successful |

**Add student page**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On load of page | It displays input fields to enter the student details. | Successful |
| 2 | Enter the values to input field | It takes all the fields as required | Successful |
| 3 | Enter the value to usn field | Checks and display the student already or not | Successful |
| 4 | On click on add button | It displays prompt messages for users and adds students to the database. | Successful |
| 5 | If any mandatory fields left empty | It displays prompt message for user | Successful |

**View student page**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On load of page | It displays all students and their details. | Successful |
| 2 | On click on view button | It will navigate to add detail view of student page | Successful |

**Add Subject page**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On load of page | It displays input fields to enter the subject details. | Successful |
| 2 | Enter the values to input field | It takes all the fields as required | Successful |
| 3 | On click on add button | It displays prompt messages for subject details added to the database. | Successful |
| 4 | If any mandatory fields left empty | It displays prompt message for user | Successful |

**Admin Students view**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI NO** | **TEST CASE** | **OBSERVED OUTPUT** | **RESULT** |
| 1 | On Load of Page | It displays the personal details and attendance of the students. | Successful |

**CHAPTER 9**

**FUTURE ENHANCEMENT**

Now in this project the admin can add, manage and view the attendance details of the student. And students can only check their attendance details, and this project is only for a single