Web Development- Minor Project!!!!!

Report On CRUD Application!!!!!!

CRUD Application-Making a Student Management System:

Minor Project-CRUD Application

- **Abstract:**

A CRUD application is a software system that enables users to create, read, update, and delete data stored in a persistent database or similar storage. It forms the backbone of many data-driven systems such as student management, inventory control, and employee records, where reliable handling of structured information is essential. In a typical CRUD application, user actions in the interface are mapped to database operations that insert new records, retrieve existing data, modify fields, or remove entries, ensuring that information stays accurate and up to date. The application usually follows a layered architecture separating presentation, business logic, and data access layers, which improves maintainability, scalability, and ease of adding features like search, filtering, and reporting. By providing standardized operations for managing data, CRUD applications simplify development, support access control and validation, and serve as a foundation for more complex enterprise and web solutions.

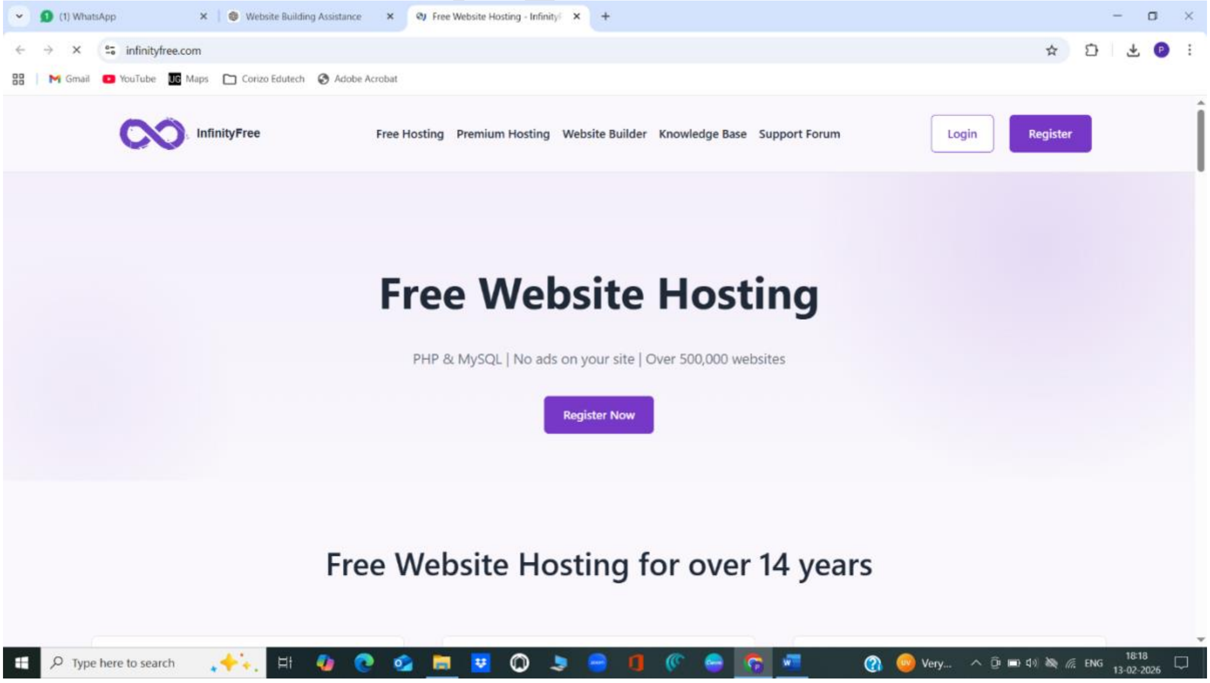
-
- CRUD stands for Create, Read, Update, and Delete, the four basic operations for managing stored data.

- CRUD applications are used in many domains, including student management, HR systems, travel booking, and e-commerce.
- User interface actions (add, view, edit, delete) are directly linked to database commands that manipulate records.
- A typical CRUD app uses a layered structure with separate UI, business logic, and data access components.
- CRUD design improves data integrity, security, and validation while simplifying future enhancements and maintenance.

ct-CRUD Application

Making a Student Management System

I took a simple and easy method to create a CRUD Application. Basically, I have used a guide as ChatGPT but I have followed the steps and learnt in detail about how the application works. I have used a website called **Infinity Free** to build it.



Free Website Hosting

PHP & MySQL | No ads on your site | Over 500,000 websites

Free Website Hosting for over 14 years

STEP 1: Creating the Database

-> Creating the Database

In order to store and manage the data for the CRUD application, a MySQL database was created using the hosting control panel provided by InfinityFree.

A database is required to store student records such as name, email, and course. Without a database, the application cannot perform Create, Read, Update, and Delete operations.

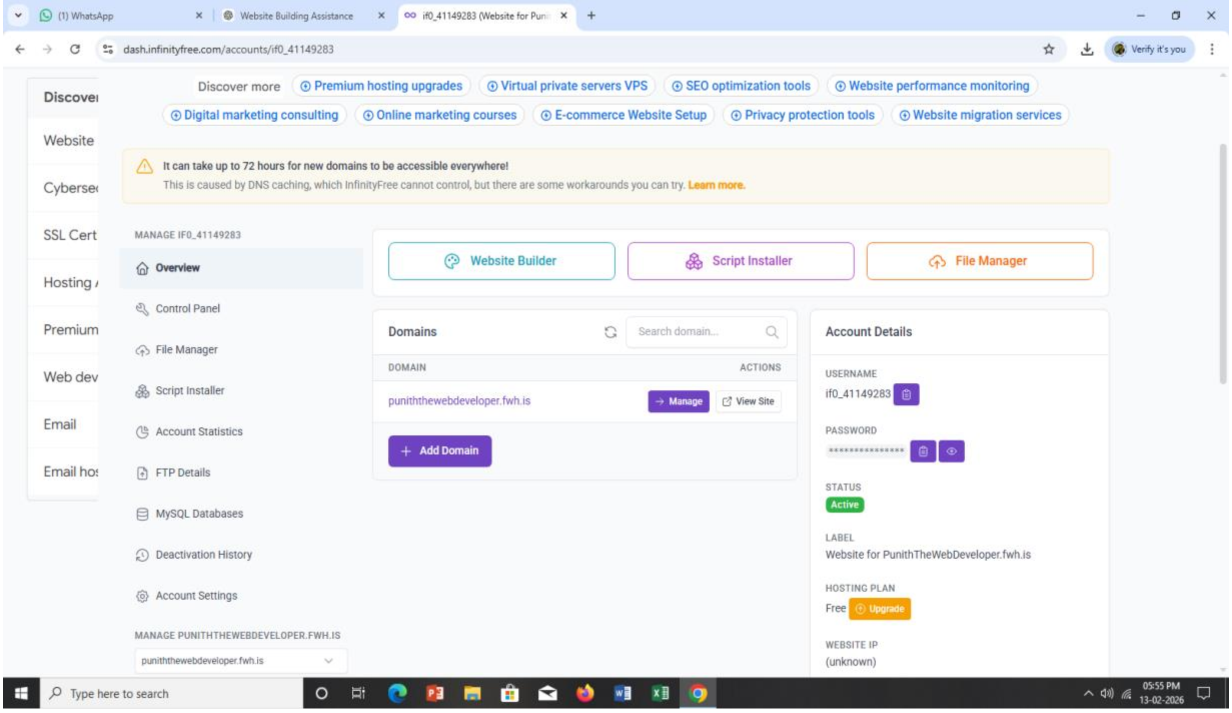
CRUD Application

-> Accessing the Hosting Control Panel

The following steps were followed:

1. Opened a web browser.
2. Logged into the Infinity Free hosting account.
3. Navigated to the **Control Panel** section.
4. Selected the option **MySQL Databases**.

This section allows users to create and manage databases.



It can take up to 72 hours for new domains to be accessible everywhere!
This is caused by DNS caching, which InfinityFree cannot control, but there are some workarounds you can try. Learn more.

Domains

DOMAIN	ACTIONS
puniththewebdeveloper.fwh.is	Manage View Site

Account Details

USERNAME
if0_41149283

PASSWORD
****

STATUS
Active

LABEL
Website for PunithTheWebDeveloper.fwh.is

HOSTING PLAN
Free

WEBSITE IP
(unknown)

->Creating a New Database

To create a new database, the following procedure was followed:

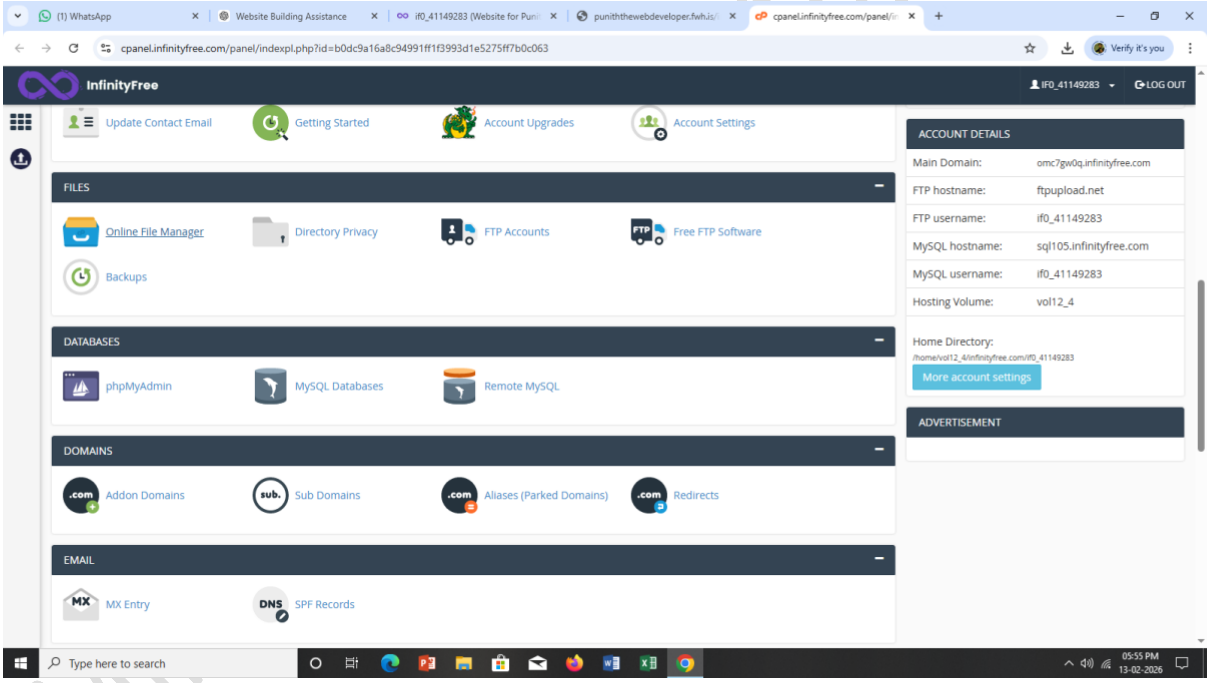
1. Clicked on **Create Database**.
2. Entered a database name (for example: crudproject).
3. Clicked on **Create** button.

After creation, the system generated:

- Database Name (example: epiz_123456_crudproject)
- Database Username
- Database Password
- Host Name (example: sql312.infinityfree.com)

These credentials are very important because they are used in the PHP file to establish a connection with the database.

The Credentials Cannot be displayed about as it's a personal data.



ACCOUNT DETAILS

Main Domain:	omc7gw0q.infinityfree.com
FTP hostname:	ftpupload.net
FTP username:	if0_41149283
MySQL hostname:	sql105.infinityfree.com
MySQL username:	if0_41149283
Hosting Volume:	vol12_4

Home Directory:
/home/vol12_4/infinityfree.com/if0_41149283

->Opening phpMyAdmin

After creating the database:

1. Clicked on **phpMyAdmin** option in the Control Panel.
2. Selected the newly created database from the left panel.
3. Clicked on the **SQL** tab.

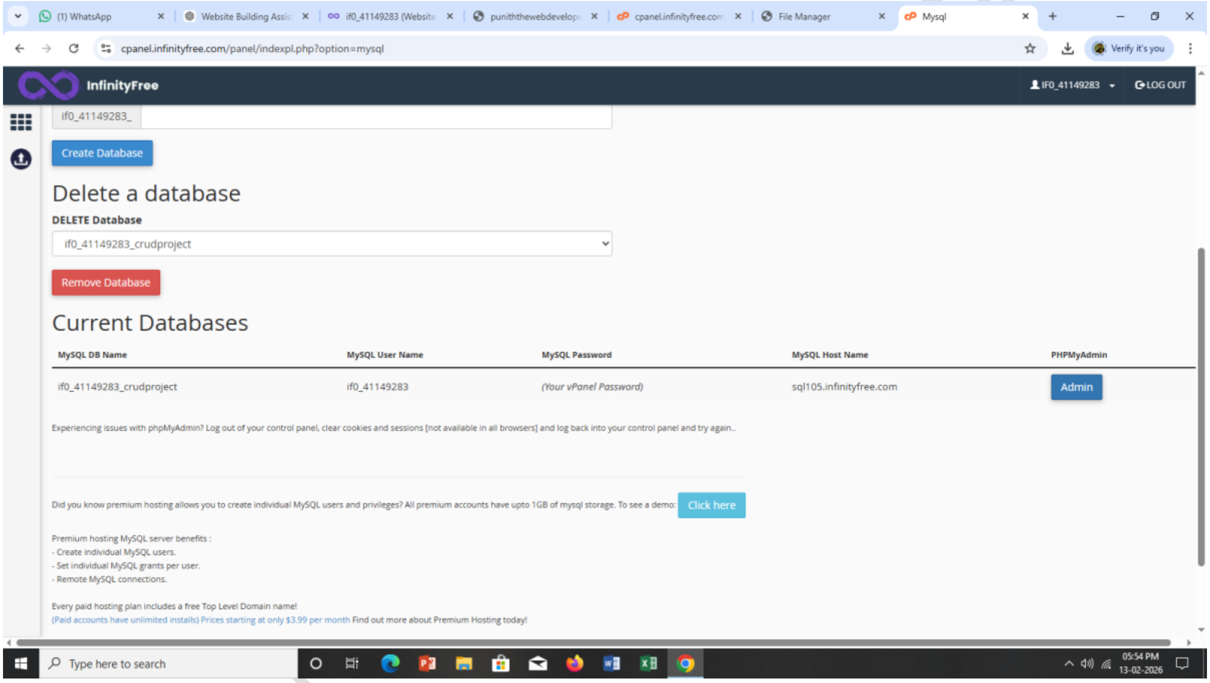
phpMyAdmin is a tool used to manage MySQL databases through a graphical interface.

Mini Application

Thus, I have used an online method to do the CRUD Application

We will use:

- **Online PHP + MySQL Hosting**
- Browser only
- No software installation



cpanel.infinityfree.com/panel/indexpl.php?option=mysql

if0_41149283_

Create Database

Delete a database

DELETE Database

if0_41149283_crudproject

Remove Database

Current Databases

MySQL DB Name	MySQL User Name	MySQL Password	MySQL Host Name	PHPMyAdmin
if0_41149283_crudproject	if0_41149283	*(Your vPanel Password)*	sql105.infinityfree.com	Admin

Experiencing issues with phpMyAdmin? Log out of your control panel, clear cookies and sessions [not available in all browsers] and log back into your control panel and try again..

Did you know premium hosting allows you to create individual MySQL users and privileges? All premium accounts have upto 1GB of mysql storage. To see a demo: Click here

Premium hosting MySQL server benefits :
- Create individual MySQL users.
- Set individual MySQL grants per user.
- Remote MySQL connections.

Every paid hosting plan includes a free Top Level Domain name!
(Paid accounts have unlimited installs) Prices starting at only $3.99 per month Find out more about Premium Hosting today!

Minor
ation

Step 2: Uploading PHP Files to the Server

After successfully creating the database and table, the next step was to upload the PHP project files to the hosting server. These files contain the logic required to perform CRUD operations.

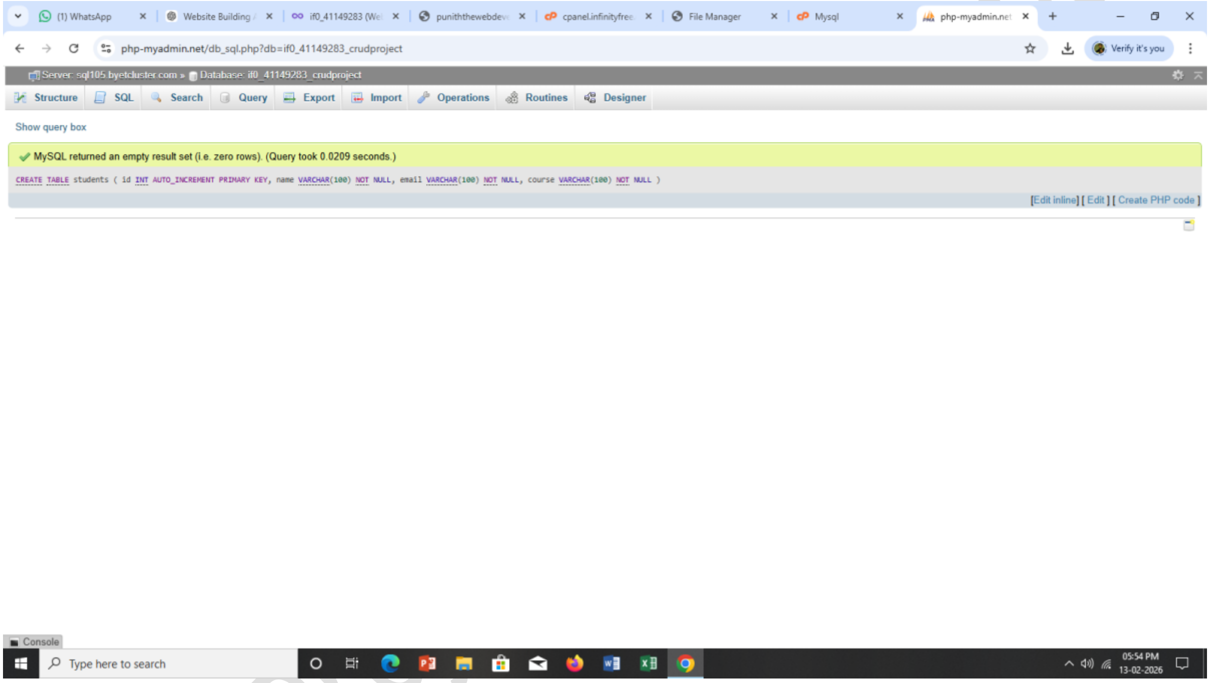
Uploading the files correctly ensures that the website runs online and connects properly to the database.

->Accessing File Manager

The following steps were followed:

1. Logged into the InfinityFree hosting account.
2. Opened the **Control Panel**.
3. Clicked on **File Manager**.
4. Opened the main directory named **htdocs**.

The htdocs folder is the root directory where all website files must be placed.



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0209 seconds.)

```
CREATE TABLE students ( id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, email VARCHAR(100) NOT NULL, course VARCHAR(100) NOT NULL )
```

Minor Pro... tion

->Creating Project Files

Inside the htdocs folder, the following files were created:

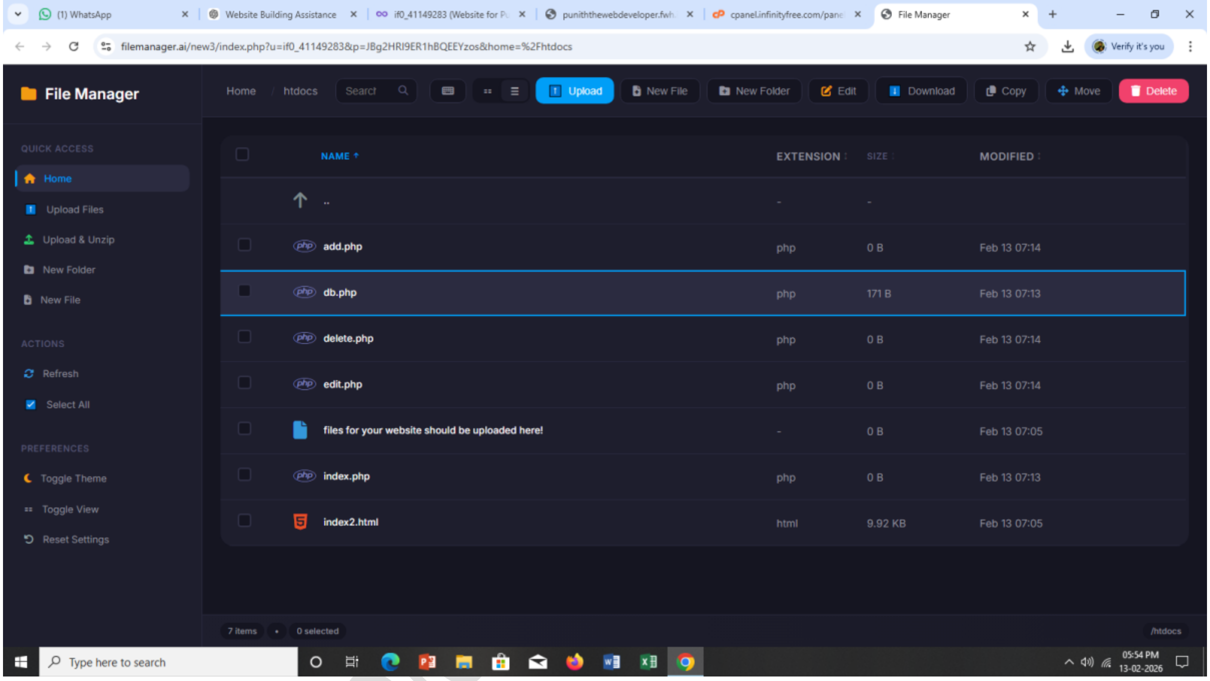
- db.php
- index.php
- add.php
- edit.php
- delete.php

Each file has a specific role in the CRUD application:

- **db.php** – Connects PHP to MySQL database
- **index.php** – Displays all records (Read operation)
- **add.php** – Adds new records (Create operation)

- **edit.php** – Updates existing records (Update operation)
- **delete.php** – Deletes records (Delete operation)

These are the need options in a CRUD Applications that is mentioned in the minor project list thus nearing to completion of the website interface. I have made simple and didn't use any bootstrap software for making the website better looking.



NAME	EXTENSION	SIZE	MODIFIED
..	-	-	
add.php	php	0 B	Feb 13 07:14
db.php	php	171 B	Feb 13 07:13
delete.php	php	0 B	Feb 13 07:14
edit.php	php	0 B	Feb 13 07:14
files for your website should be uploaded here!	-	0 B	Feb 13 07:05
index.php	php	0 B	Feb 13 07:13
index2.html	html	9.92 KB	Feb 13 07:05

filemanager.ai/new3/index.php?u=if0_41149283&p=JBg2HRI9ER1hBQEEYzos&home=%2Fhtdocs

7 items • 0 selected /htdocs

The Code used for all the php files are given below: -

->db.php

```
<?php
$conn = mysqli_connect("YOUR_HOST",
"YOUR_USERNAME", "YOUR_PASSWORD",
"YOUR_DATABASE");

if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}
?>
```

->index.php

Minor Project

```
<?php include 'db.php'; ?>
<h2>Student Management System</h2>
<a href="add.php">Add Student</a>
<table border="1">
<tr>
<th>ID</th>
<th>Name</th>
<th>Email</th>
<th>Course</th>
<th>Action</th>
</tr>

<?php
$result = mysqli_query($conn, "SELECT * FROM
students");
while($row = mysqli_fetch_assoc($result)){
?>
<tr>
<td><?php echo $row['id']; ?></td>
<td><?php echo $row['name']; ?></td>
<td><?php echo $row['email']; ?></td>
<td><?php echo $row['course']; ?></td>
<td>
<a href="edit.php?id=<?php echo $row['id']; ?>">Edit</a>
<a href="delete.php?id=<?php echo
$row['id']; ?>">Delete</a>
</td>
</tr>
<?php } ?>
</table>
```

->add.php

```
<?php
include 'db.php';
```

Minor Project-CRUD Application

```
if(isset($_POST['submit'])){
$name = $_POST['name'];
$email = $_POST['email'];
$course = $_POST['course'];

mysqli_query($conn,"INSERT INTO
students(name,email,course)
VALUES('$name','$email','$course')");

header("Location:index.php");
}
?>

<form method="POST">
Name: <input type="text" name="name"
required><br>
Email: <input type="email" name="email"
required><br>
Course: <input type="text" name="course"
required><br>
<button name="submit">Add</button>
</form>
```

->edit.php

```
<?php
include 'db.php';
$id = $_GET['id'];
$result = mysqli_query($conn,"SELECT * FROM students
WHERE id=$id");
$row = mysqli_fetch_assoc($result);

if(isset($_POST['update'])){
$name = $_POST['name'];
```

Minor Project-CRUD Application

```
$email = $_POST['email'];
$course = $_POST['course'];

mysqli_query($conn,"UPDATE students SET
name='$name',
email='$email',
course='$course'
WHERE id=$id");

header("Location:index.php");
}
?>

<form method="POST">
Name: <input type="text" name="name" value="<?php echo
$row['name']; ?>"><br>
Email: <input type="email" name="email" value="<?php echo
$row['email']; ?>"><br>
Course: <input type="text" name="course" value="<?php echo
$row['course']; ?>"><br>
<button name="update">Update</button>
</form>
```

->delete.php

```
<?php
include 'db.php';
$id = $_GET['id'];
mysqli_query($conn,"DELETE FROM students WHERE
id=$id");
header("Location:index.php");
?>
```

Minor Project-CRUD Application

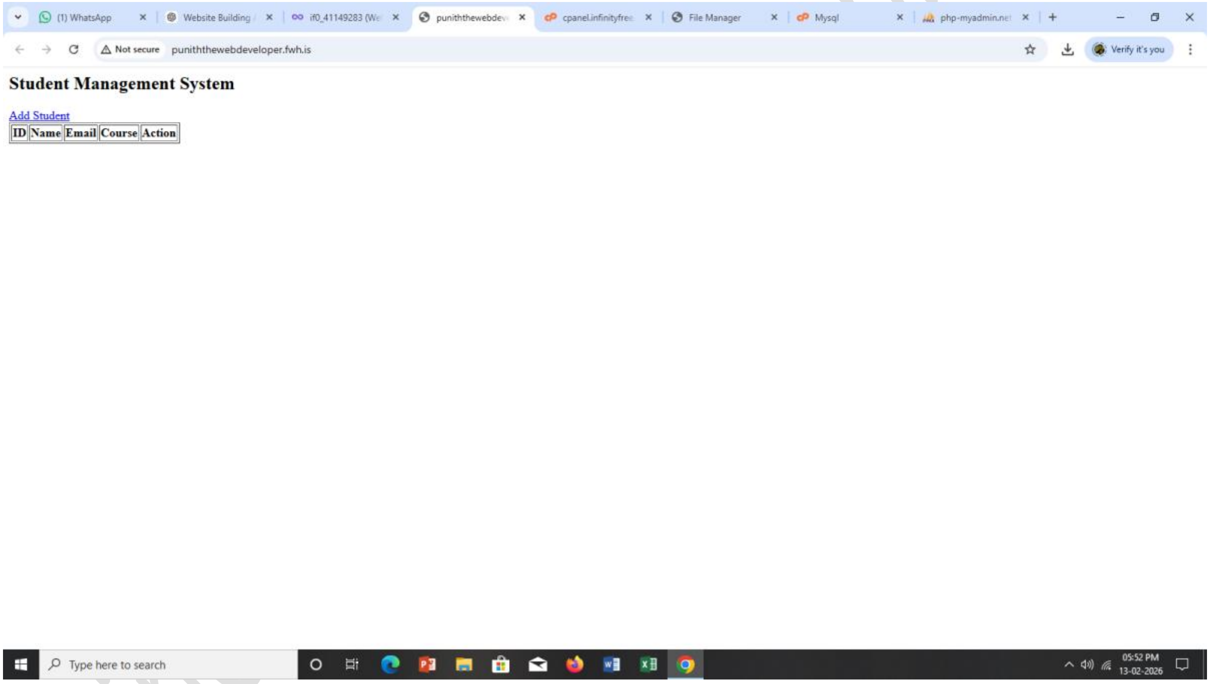
Technologies Used

- Frontend: HTML, CSS
- Backend: PHP
- Database: MySQL
- Hosting Platform: InfinityFree
- Database Tool: phpMyAdmin

System Requirements

- Web Browser (Chrome/Edge)
- Internet Connection
- InfinityFree Hosting Account

plication



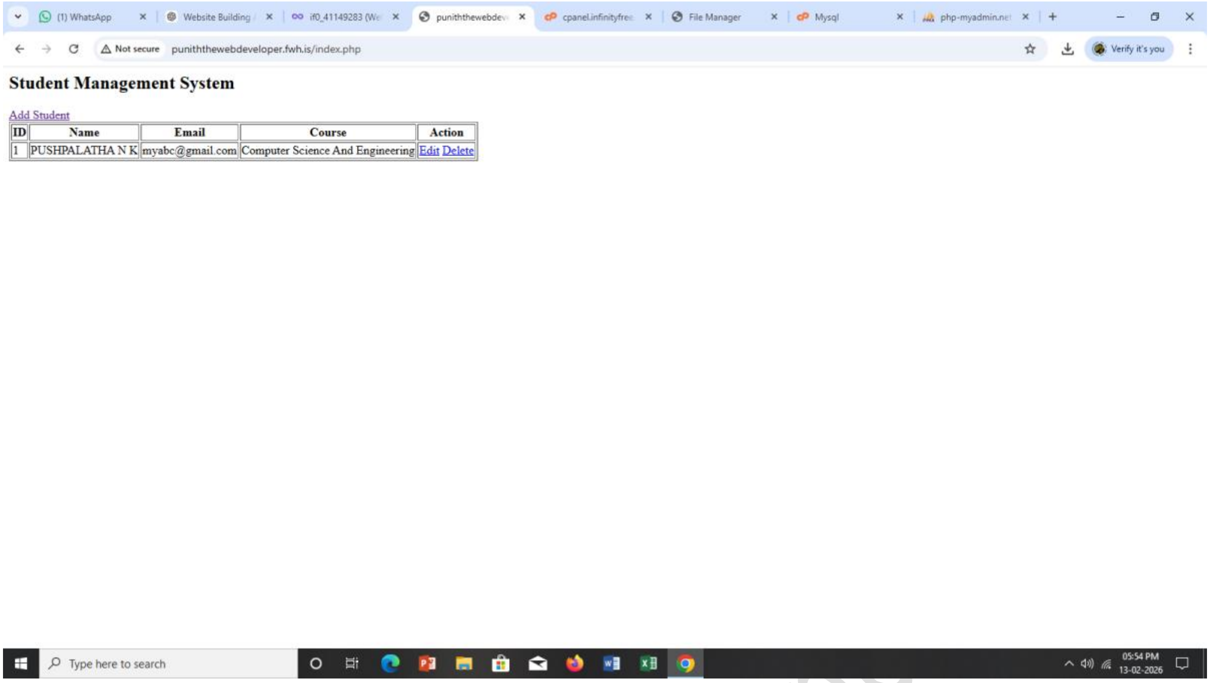
Student Management System

Add Student

ID	Name	Email	Course	Action

Min

After Adding a Student Data



Student Management System

Add Student

ID	Name	Email	Course	Action
1	PUSHPALATHA N K	myabc@gmail.com	Computer Science And Engineering	Edit Delete

->Results

The CRUD Application was successfully developed and deployed online. All CRUD operations worked correctly, and the project demonstrated real-time database interaction.

-> Conclusion

The CRUD Application helped in understanding:

- Database creation
- Server-side scripting
- Data manipulation using SQL queries
- Hosting a website online

This project provides practical knowledge of web development concepts and database management.

-> Future Enhancements

- Add login authentication system
- Improve UI using Bootstrap
- Add search functionality
- Add validation and security improvements

Minor Project-CRUD App

Testing CRUD Operations

To verify proper functionality, all CRUD operations were tested:

Create Operation

- Clicked “Add Student”.
- Entered name, email, and course.
- Clicked Submit.
- Record was successfully inserted into the database.

Read Operation

- After adding data, all student records were displayed in table format.
- Verified that data appears correctly.

Update Operation

- Clicked Edit button next to a student record.
- Modified the details.
- Clicked Update.
- Verified that changes were saved successfully.

Delete Operation

- Clicked Delete button.
- Confirmed deletion.
- Record was removed from the database.

Minor Project-CRUD Application

Opening the Website

The following procedure was followed:

1. Opened a web browser (Google Chrome).
2. Entered the website URL provided by InfinityFree.

Example format:

https://yourproject.infinityfreeapp.com

3. Pressed Enter.

The homepage (index.php) loaded successfully.

In the photos uploaded above you can see the website you can press and see if it's working.

->Final Verification

After testing all operations:

- No database connection errors were found.
- All CRUD operations worked successfully.
- Website was accessible online.

This confirmed that the project was successfully deployed and functioning properly.

The PHP files were successfully uploaded to the hosting server, and the database connection was properly configured. The project was executed online and all CRUD operations (Create, Read, Update, Delete) worked correctly without errors.

Thus, I end my minor project report by this note.

Punith.C
(Student of BMS College of Engineering)

Thank you.

Minor Project-CRUD Application