

CampusConnect – Smart Campus Utility Platform

*A Mini Project Report Submitted in Partial Fulfilment of
Master of Computer Applications – Web Application Development*

Submitted By:

Roll No. SIC No. Name

-- -- **ADITYA PRAKASH**

-- -- **ASHUTOSH DASH**

-- -- **RITESH SINGH**

Submitted To:

[Faculty Name]

University Name:

[University Name]

Department:

Master of Computer Applications

Academic Year:

2025–26

Table of Contents

S.No	Contents	Page No.
1	Introduction	03
2	Proposed System	04–05
3	System Design	06–07
4	Implementation	08–16
5	Screenshots	17–21
6	Conclusion	22
7	References	22

1. INTRODUCTION

The modern university campus is a complex ecosystem where students must balance academic responsibilities with social engagement and daily living needs. Traditionally, these aspects are managed through fragmented, manual, or disconnected systems—leading to inefficiencies and a suboptimal campus experience.

CampusConnect is a unified, web-based solution that centralizes essential student activities. It integrates four core modules:

- **Note Sharing Repository**
- **Real-time Chat System**
- **Food Ordering System**
- **Rent/Accommodation Finder**

The application is built using the **Laravel PHP Framework (v10)** following the MVC architecture. Real-time chat features are powered by **Laravel Broadcasting & Pusher**, and **MySQL** is used for secure database management. The frontend uses **Blade Templates, HTML5, CSS3, and Bootstrap 5**.

CampusConnect aims to simplify campus life by enabling students to access study materials, order food, find accommodation, and communicate effectively—all through a single platform.

2. PROPOSED SYSTEM

The proposed system centralizes multiple campus-related functionalities into one platform. It also introduces a **credit-based gamification system** where students earn credits for contributing content.

Objectives

1. Provide a unified platform for academic and non-academic campus activities.
2. Facilitate peer-to-peer academic support via a note-sharing repository.
3. Enable real-time communication through chat rooms.
4. Streamline food ordering to reduce waiting time.
5. Assist students in discovering accommodation listings.
6. Implement role-based access for Students and Administrators.
7. Ensure data security through modern web practices.
8. Encourage student engagement via a credit system.
9. Provide a fully responsive UI across all devices.
10. Simplify administrative tasks such as approvals and content management.

Module-Wise Functionality

Student Module

Functionality	Description
----------------------	--------------------

Login/Register	Secure authentication system.
----------------	-------------------------------

Note Sharing	Upload/view notes, earn credits for approved uploads.
--------------	---

Chat Room	Participate in public/private real-time chat rooms.
-----------	---

Food Ordering	Browse menu, order food, track status.
---------------	--

Rent/PG Finder	Browse/post property listings, earn credits.
----------------	--

Admin Module

Functionality	Description
----------------------	--------------------

Dashboard	Overview of system activity.
-----------	------------------------------

Manage Notes	Approve/reject uploaded notes.
--------------	--------------------------------

Manage Food Orders	Update food order statuses.
--------------------	-----------------------------

Manage Listings	Moderate accommodation posts.
-----------------	-------------------------------

User Management	Manage student accounts and roles.
-----------------	------------------------------------

3. SYSTEM DESIGN

Database Table Structure

users

- id
- name
- email
- password
- role (default: student)
- credits (default: 0)
- timestamps

notes

- id
- user_id
- title
- department
- semester
- subject
- file_name
- status (pending/approved/rejected)
- timestamps

rents

- id
- title
- rent
- location
- amenities
- owner_name

- phone
- created_by
- availability_status
- approved (boolean)
- timestamps

food_orders

- id
- user_id
- restaurant_id
- total_amount
- credits_used
- status (received/preparing/ready)
- timestamps

chat_rooms

- id
- room_name
- type (public/private)
- created_by
- timestamps

Use-Case Diagram (Description)

Student Use Cases

- Login/Register
- Upload/Search/Download Notes
- Join Chat Room
- Send Messages
- Browse Food Menu
- Place Food Order
- View/Post Rent Listings

Admin Use Cases

- Login
- Approve/Reject Notes
- Manage Food Orders
- Manage Rent Listings
- View Dashboard Statistics

4. IMPLEMENTATION

Technology Stack

- **HTML5**
- **CSS3**
- **Bootstrap 5**
- **JavaScript**
- **PHP 8.1+**
- **Laravel 10**
- **MySQL**
- **Pusher (WebSockets)**

Sample Code

1. Home.blade.php

```
@extends('layouts.app')

@section('content')
<div class="text-center py-5">
    <div class="mb-5">
        <h1 class="display-3 fw-bold mb-3">
            Welcome to <span class="text-primary-gradient">CampusConnect</span> 🙌
        </h1>
        <p class="lead text-muted mb-4 mx-auto" style="max-width: 600px;">
            Your all-in-one student platform. Access notes, join chat rooms, order food, and find
            the perfect place to stay.
        </p>
    </div>
</div>

@auth
    <div class="glass-card d-inline-block px-4 py-2 mb-4">
        <p class="mb-0 text-muted small">Logged in as</p>
        <p class="mb-0 fw-bold">{{ auth()->user()->name }}</p>
        <div class="mt-2 badge bg-danger bg-opacity-10 text-danger border border-danger
border-opacity-25">
            Credits: {{ auth()->user()->credit_balance }}
        </div>
    </div>
</div>
@endauth
</div>
```

```

<div class="row g-4 justify-content-center">
    <div class="col-md-6 col-lg-3">
        <div class="glass-card p-4 h-100 d-flex flex-column align-items-center transition-hover">
            <h5 class="fw-bold mb-2">Notes Section</h5>
            <p class="text-muted small mb-4">Upload and share study materials with peers.</p>
            <a href="{{ route('notes.view') }}" class="btn btn-outline-custom w-100 mt-auto">Browse Notes</a>
        </div>
    </div>

    <div class="col-md-6 col-lg-3">
        <div class="glass-card p-4 h-100 d-flex flex-column align-items-center transition-hover">
            <h5 class="fw-bold mb-2">Chat Room</h5>
            <p class="text-muted small mb-4">Connect with students in public or private rooms.</p>
            <a href="{{ route('chat.index') }}" class="btn btn-outline-custom w-100 mt-auto">Join Chat</a>
        </div>
    </div>
</div>
@endsection

```

2. Register.blade.php

```

@extends('layouts.guest')
@section('title', 'Register')

```

```

@section('content')

<div class="auth-wrapper">

    <div class="login-card text-center">
        <div class="logo-circle">CC</div>
        <h2 class="mt-3">CampusConnect</h2>

        <form action="{{ route('register.perform') }}" method="POST" class="mt-4">
            @csrf
            <input type="text" name="name" placeholder="Name" class="form-control mb-3" required>
            <input type="email" name="email" placeholder="Email" class="form-control mb-3" required>
            <input type="password" name="password" placeholder="Password" class="form-control mb-3" required>
            <button class="btn-main" type="submit">Register</button>
        </form>

        <a href="{{ route('login') }}" class="d-block mt-3">Already have an account? Login</a>
    </div>
</div>

@endsection

```

3. Login.blade.php

```

@extends('layouts.guest')

@section('title','Login')

@section('content')
<div class="auth-wrapper">
    <div class="login-card text-center">

```

```

<div class="logo-circle">CC</div>

<h2 class="mt-3">CampusConnect</h2>

<form action="{{ route('login.perform') }}" method="POST" class="mt-4">
    @csrf

    <input type="email" name="email" placeholder="Email" class="form-control mb-3" required>

    <input type="password" name="password" placeholder="Password" class="form-control mb-3" required>

    <button class="btn-main" type="submit">Log In</button>

</form>

<a href="{{ route('register') }}" class="d-block mt-3">Don't have an account?  

Register</a>

</div>
</div>
@endsection

```

4. NotesController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Note;
use Illuminate\Support\Facades\Auth;

class NotesController extends Controller
{
    public function index()

```

```

{

$notes = Note::where('status','approved')->orderBy('created_at','desc')->get();

return view('notes.view', compact('notes'));

}

public function upload(Request $request)

{

$request->validate([

'title' => 'required|string|max:255',

'department' => 'required|string|max:255',

'semester' => 'required|string|max:255',

'subject' => 'required|string|max:255',

'file' => 'required|mimes:pdf|max:20480',

]);


$file = $request->file('file');

$fileName = time().'.'.preg_replace('/\s+/','_',$file->getClientOriginalName());

$file->move(public_path('uploads'),$fileName);

Note::create([

'user_id' => Auth::id(),

'title' => $request->title,

'department' => $request->department,

'semester' => $request->semester,

'subject' => $request->subject,

'file_name' => $fileName,

'status' => 'pending',


]);
}

```

```
    return back()->with('success','Notes uploaded successfully!');

}

}
```

5. SCREENSHOTS

(Place your screenshots here)

- Home Page
- Registration Page
- Login Page
- Notes View
- Food Ordering
- Chat Room
- Rent Listings
- Admin Dashboard
- Admin Note Approval
- Order Management
- User Profile

6. CONCLUSION

CampusConnect demonstrates how a unified digital solution can enhance the university experience by integrating academic collaboration, communication, accommodation, and food services onto a single platform.

Built using modern web technologies like **Laravel, Bootstrap, MySQL, and Pusher**, the system is secure, scalable, and user-friendly. The credit-based gamification encourages participation, while modular design allows easy future expansion.

Future upgrades may include:

- Payment gateway integration
 - Mobile application
 - AI-based recommendations
-

7. REFERENCES

1. Laravel Documentation – <https://laravel.com/docs/10.x>
2. Bootstrap Documentation – <https://getbootstrap.com/docs>
3. Pusher Documentation – <https://pusher.com/docs>
4. PHP Tutorial – <https://www.w3schools.com/php/>
5. MySQL Tutorial – <https://www.w3schools.com/mysql/>