

# **CU – Event Management System**

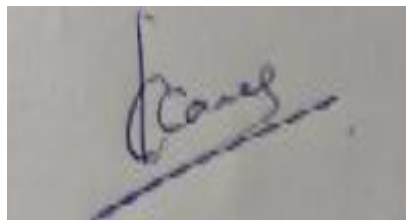
Submitted in partial fulfillment of the requirements for the award of degree of

## **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING**



**Submitted to:  
Project Teacher**  
Anjuli Goel -E7948

**Submitted By:  
Student Group**  
1. Harshit Srivastava, 19BCS1142  
2. Tanmay Tripathi, 19BCS1120



**Mentor Signature**  
MS Charanpreet Kaur, 6227

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

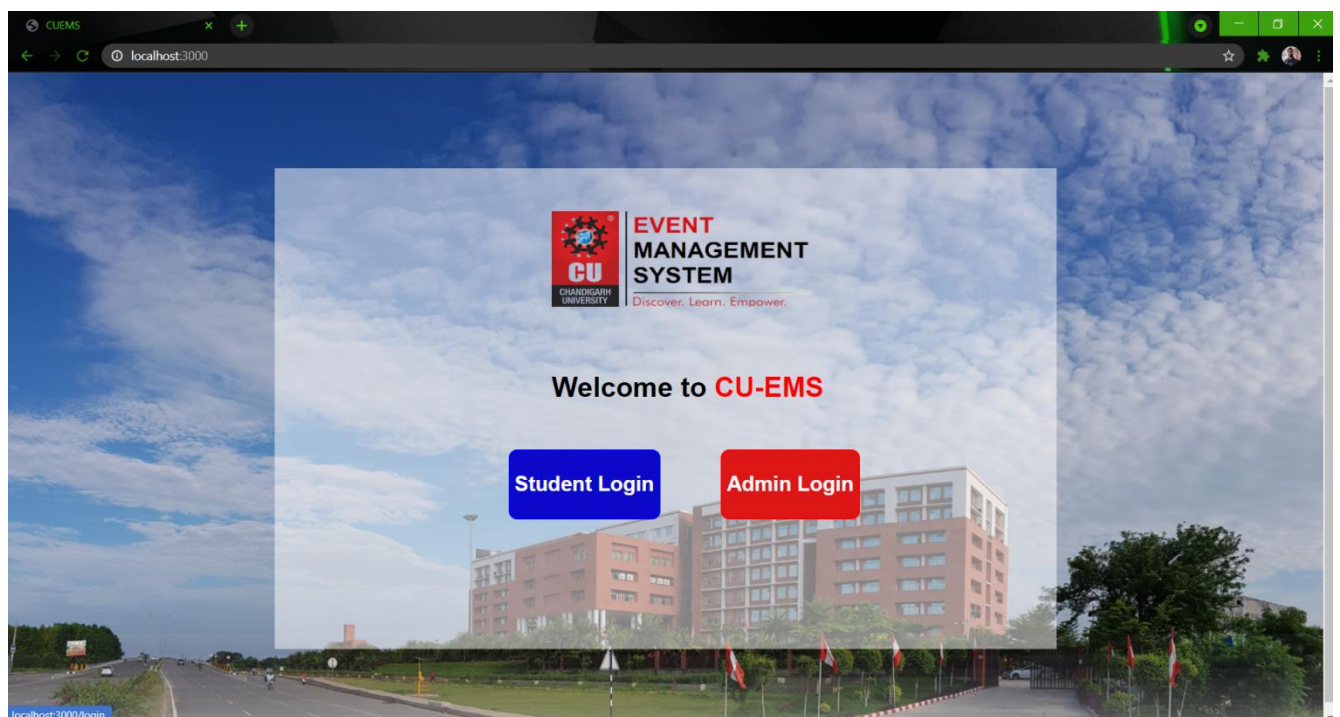
**June 2021**

# Second Progress Report

## 1. Implementation (100%)

### **1.1. Landing page :**

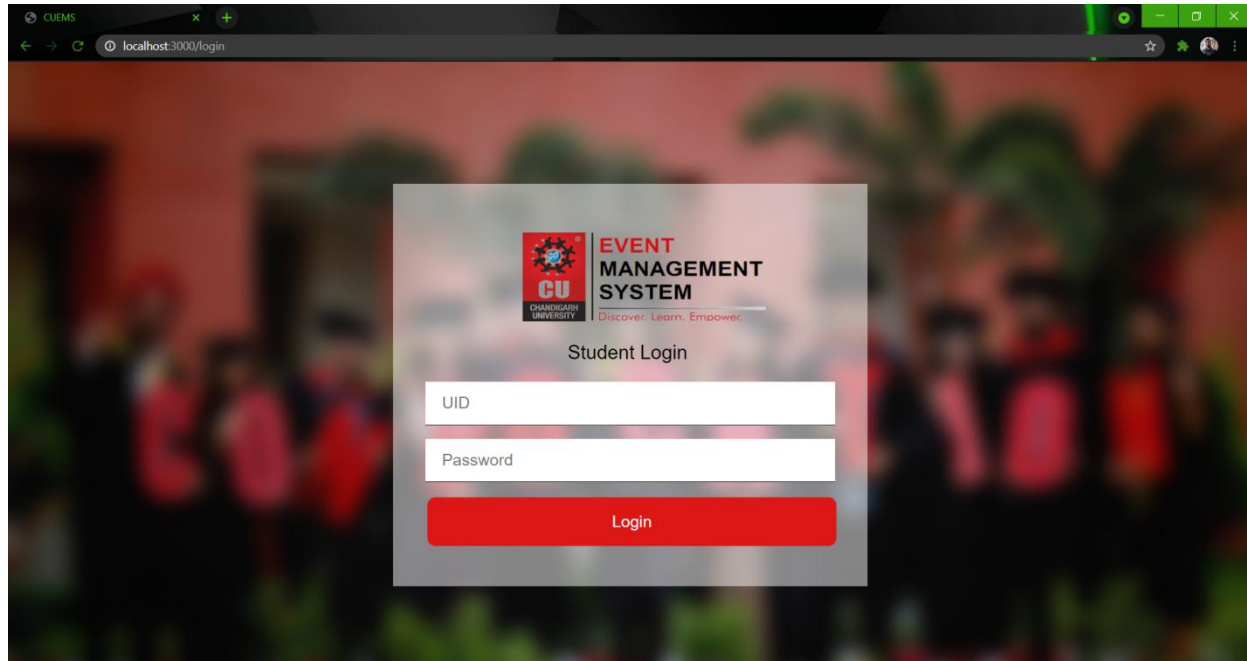
- This is the landing page of Chandigarh University – Event Management System (CU-EMS).
- In this page, there are two buttons – one for student login and other is for admin login.
- Clicking on any of the two buttons redirects you to their respective login pages.



**Figure 1 : CU-EMS Landing page**

## 1.2. Student :

### 1.2.1. Student Login page :



**Figure 2 : CU-EMS Student Login page**

### 1.2.3. Student Home page :



**Figure 3 : CU-EMS Student Homepage**

### 1.3. Admin :

#### 1.3.1. Admin Login Page :

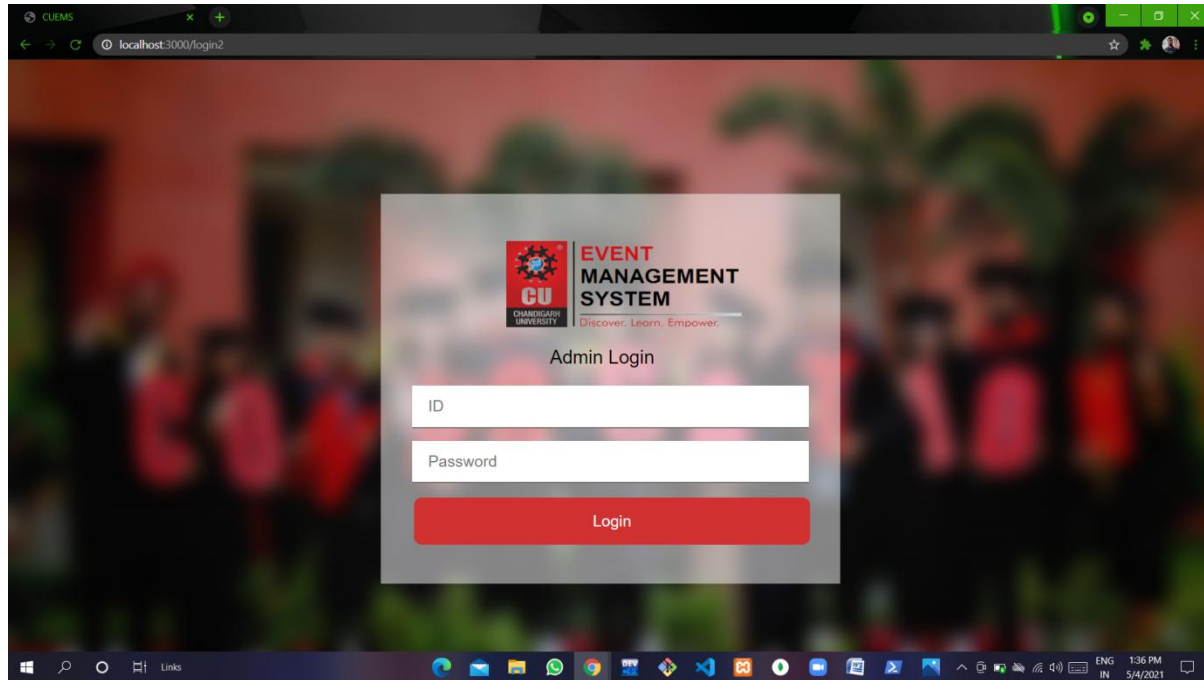


Figure 4 : CU-EMS Admin Login page

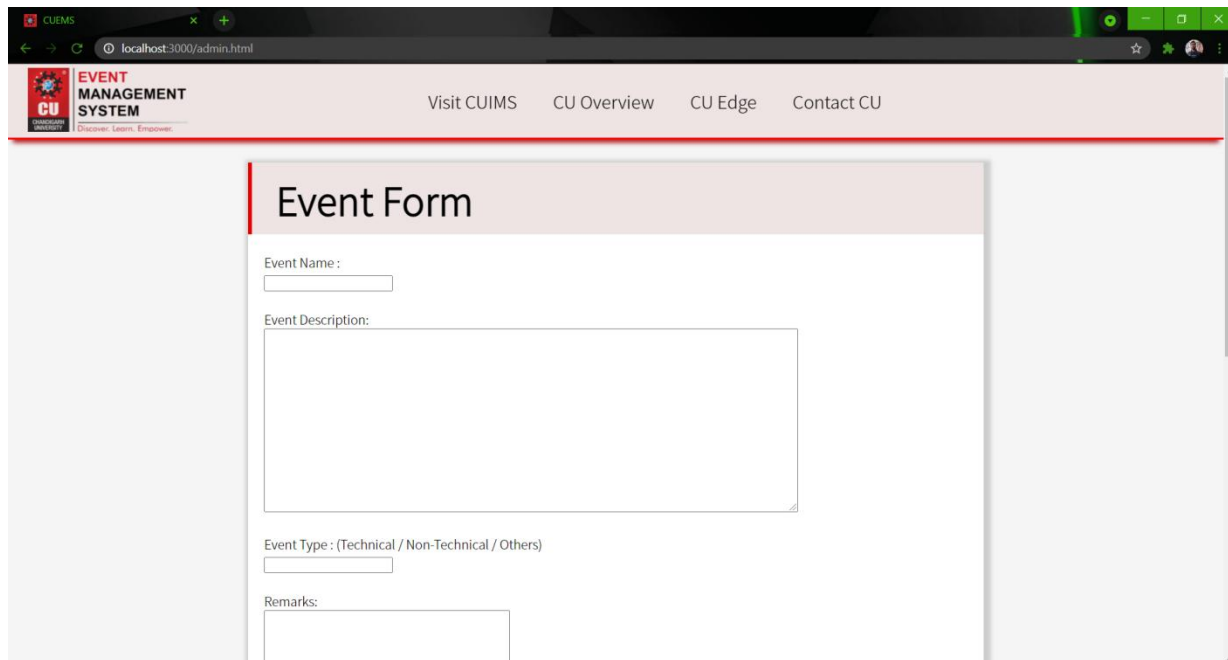
#### 1.3.2. Admin Home Page :



Figure 5 : CU-EMS Admin Homepage

### 1.3.3. Admin Event Form Page :

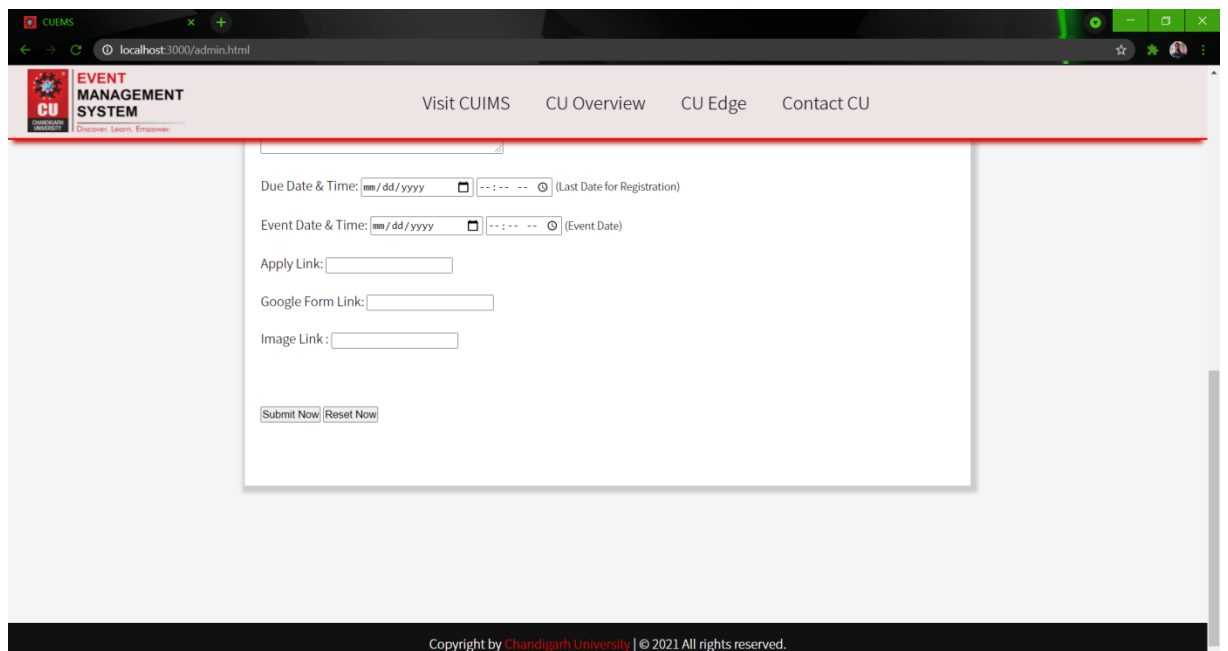
When admin clicks on the “Post” Button which is in the top-right side of the admin homepage, it redirects it to an Event form page where admin can post a new upcoming event.



The screenshot displays the 'Event Form' page in a web browser. The browser's address bar shows 'localhost:3000/admin.html'. The page header includes the 'EVENT MANAGEMENT SYSTEM' logo and navigation links: 'Visit CUIMS', 'CU Overview', 'CU Edge', and 'Contact CU'. The main content area is titled 'Event Form' and contains the following fields:

- Event Name :** A text input field.
- Event Description:** A large text area for detailed description.
- Event Type : (Technical / Non-Technical / Others)** A dropdown menu.
- Remarks:** A text input field.

**Figure 6 : CU-EMS Admin Event Form Page (1/2)**



The screenshot displays the bottom section of the 'Event Form' page. It includes the following fields and buttons:

- Due Date & Time:** A date and time picker with a calendar icon and a clock icon. The text '(Last Date for Registration)' is displayed to the right.
- Event Date & Time:** A date and time picker with a calendar icon and a clock icon. The text '(Event Date)' is displayed to the right.
- Apply Link:** A text input field.
- Google Form Link:** A text input field.
- Image Link :** A text input field.
- Submit Now** and **Reset Now** buttons.

The footer of the page contains the text: 'Copyright by Chandigarh University | © 2021 All rights reserved.'

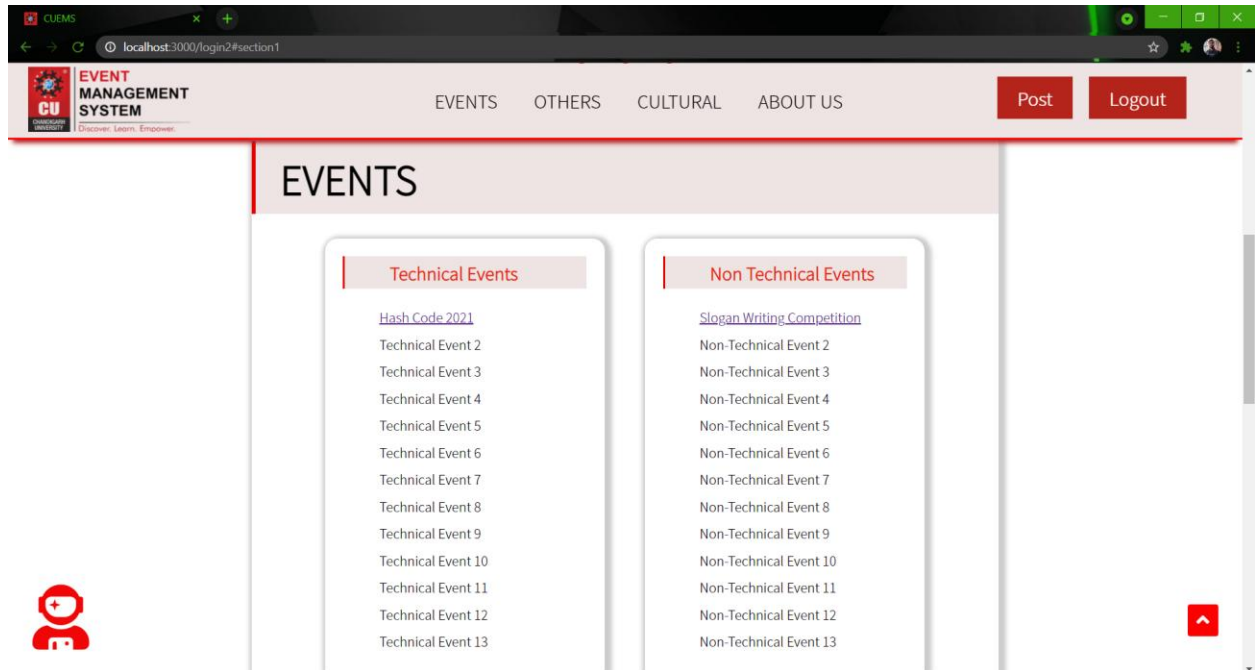
**Figure 7 : CU-EMS Admin Event Form Page (2/2)**

## 1.4. Common Homepage :

In the homepage be of admin or student there are main 3 menus in the header part – EVENTS, OTHERS, CULTURAL.

### 1.4.1. EVENTS :

When you click on the “EVENTS”, it redirects you to #section1 on the same page which can be seen below.



**Figure 8 : CU-EMS after clicking on EVENTS**



### 1.4.2. OTHERS :

When you click on the “OTHERS”, it redirects you to #section2 on the same page which can be seen below.

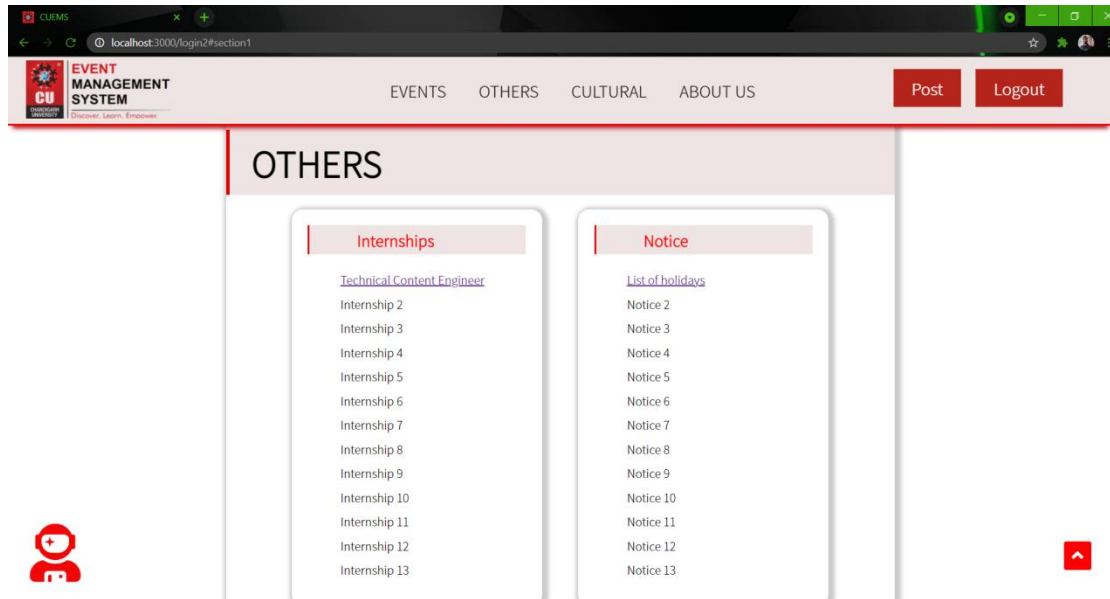


Figure 9 : CU-EMS after clicking on OTHERS

### 1.4.3. CULTURAL :

When you click on the “CULTURAL”, it redirects you to a page where all the events regarding cultural fests will be notified.

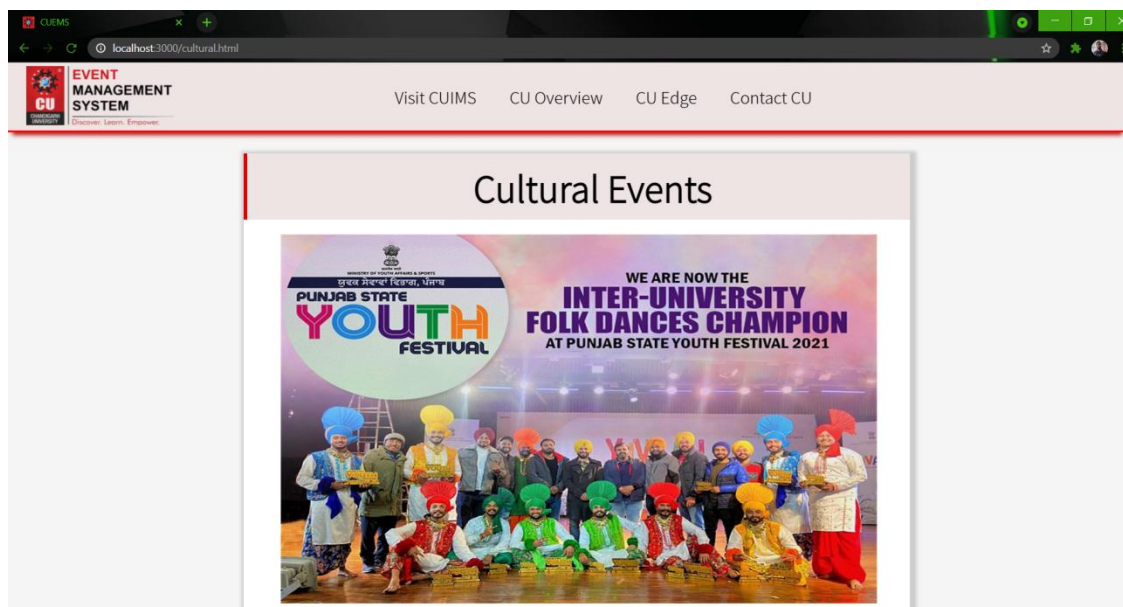
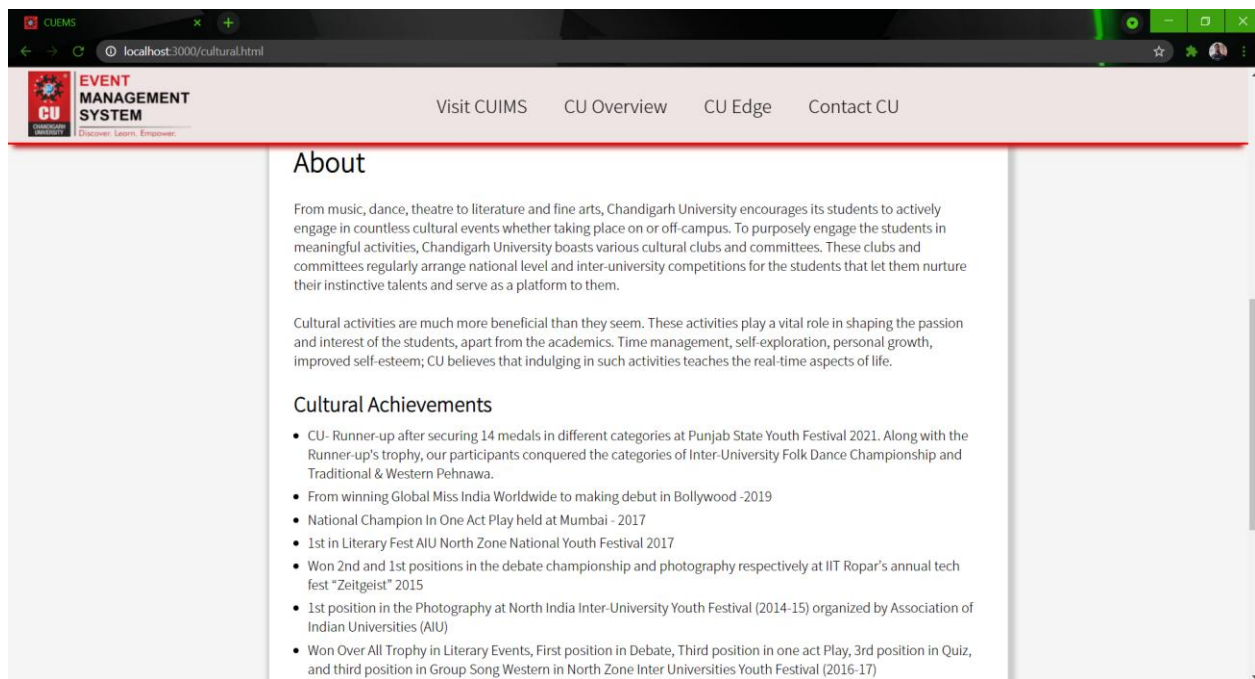
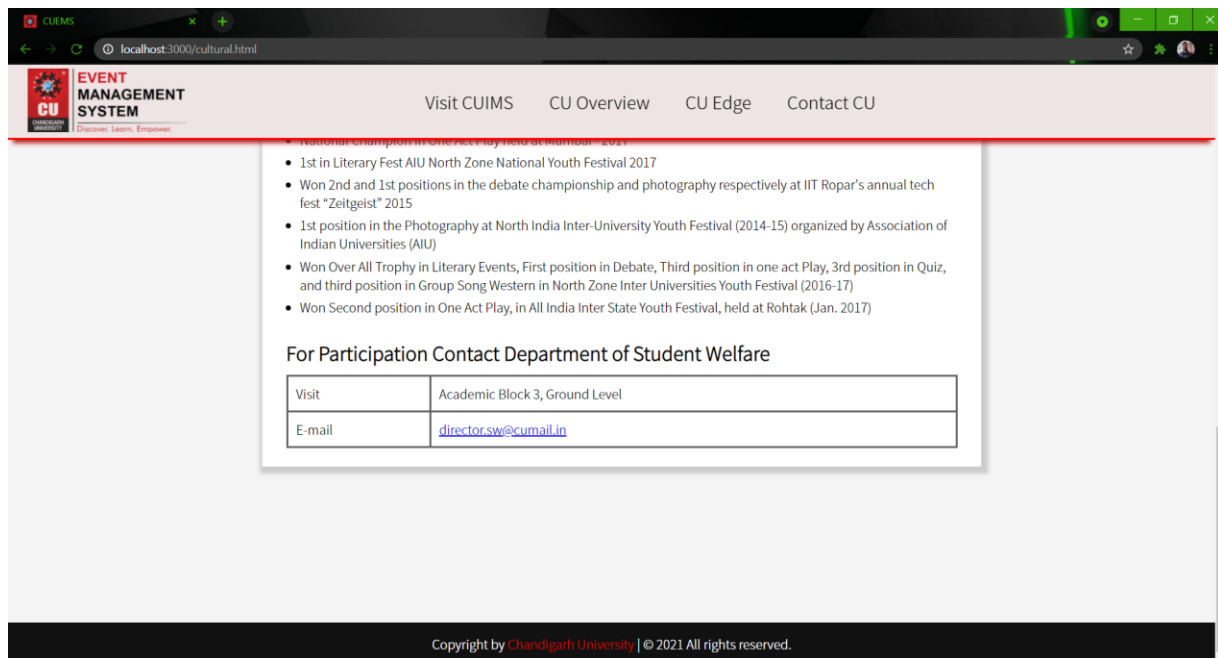


Figure 10 : CU-EMS Cultural Page (1/3)



**Figure 11 : CU-EMS Cultural Page (2/3)**



**Figure 12 : CU-EMS Cultural Page (3/3)**



## 1.5. Types of Events :

### 1.5.1. Technical Events :

Below is the sample page of a Technical Event.

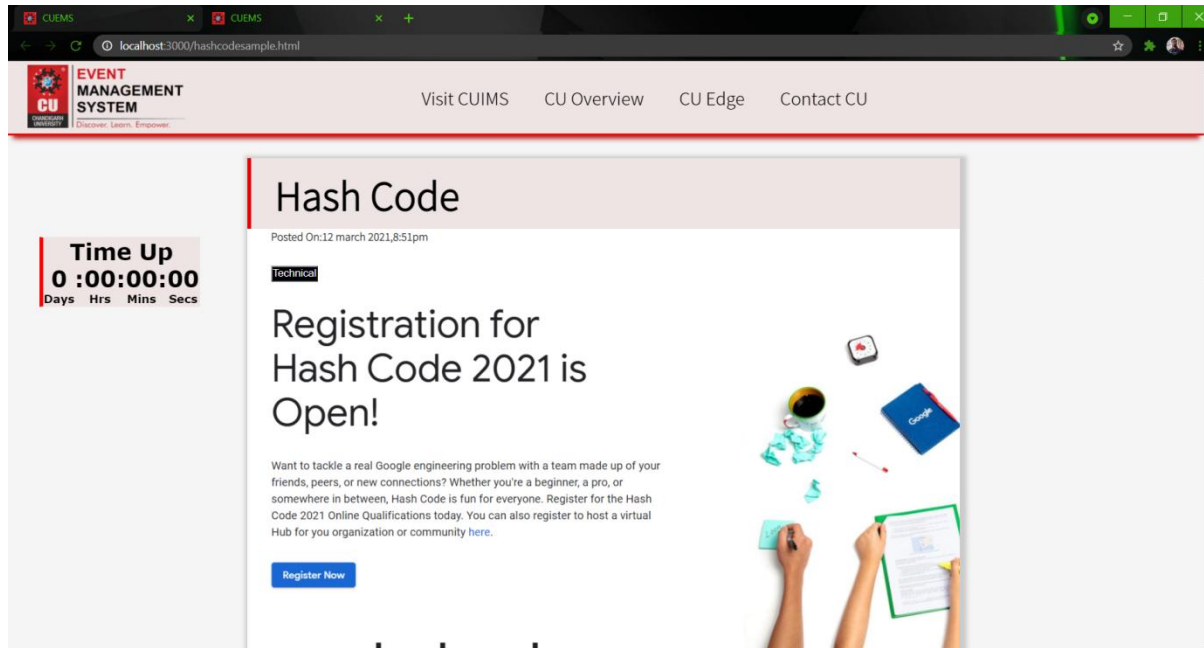


Figure 13 : CU-EMS sample technical event page (1/3)

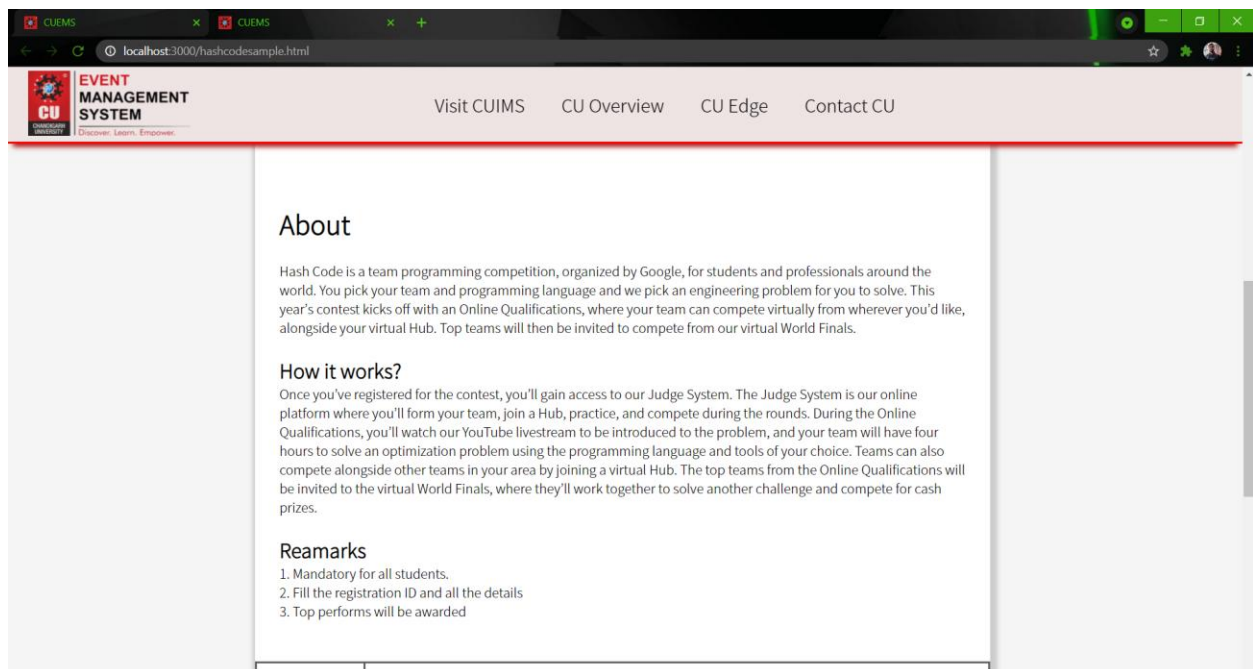
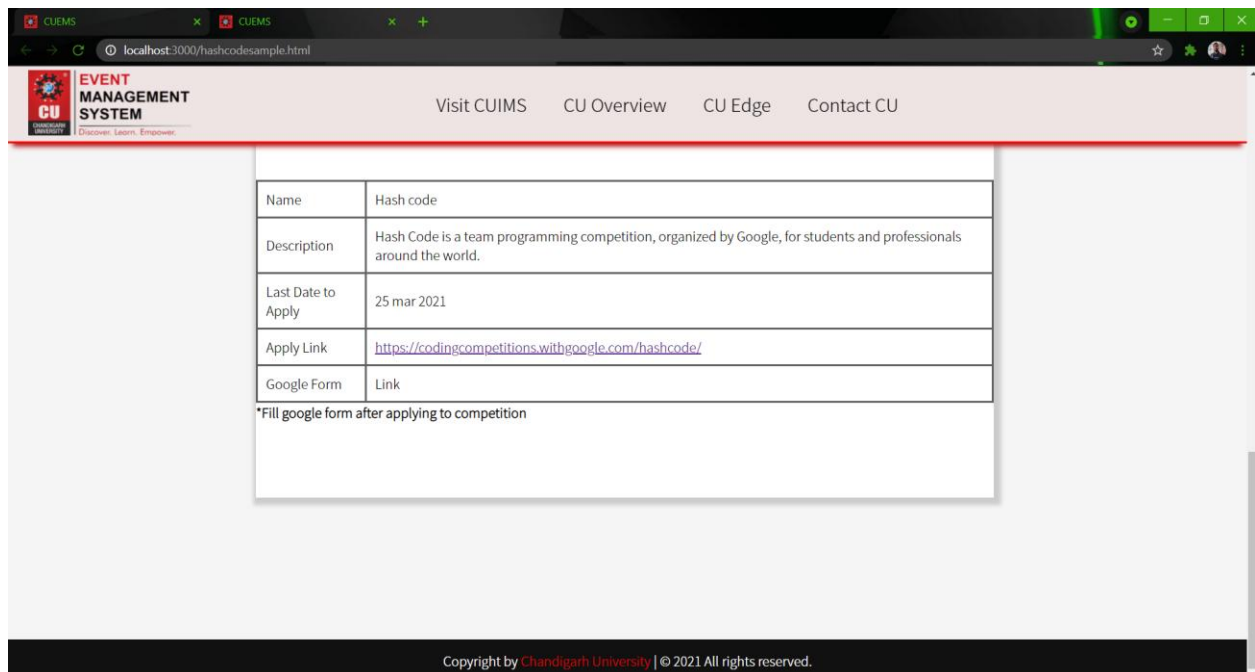


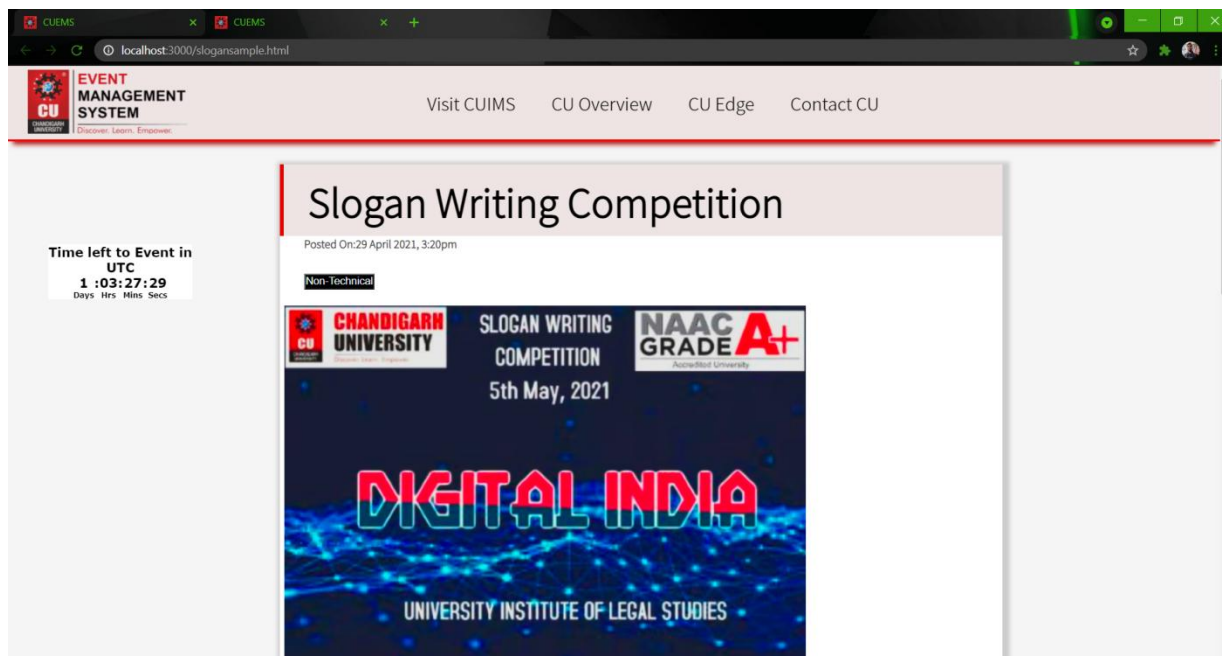
Figure 14 : CU-EMS sample technical event page (2/3)



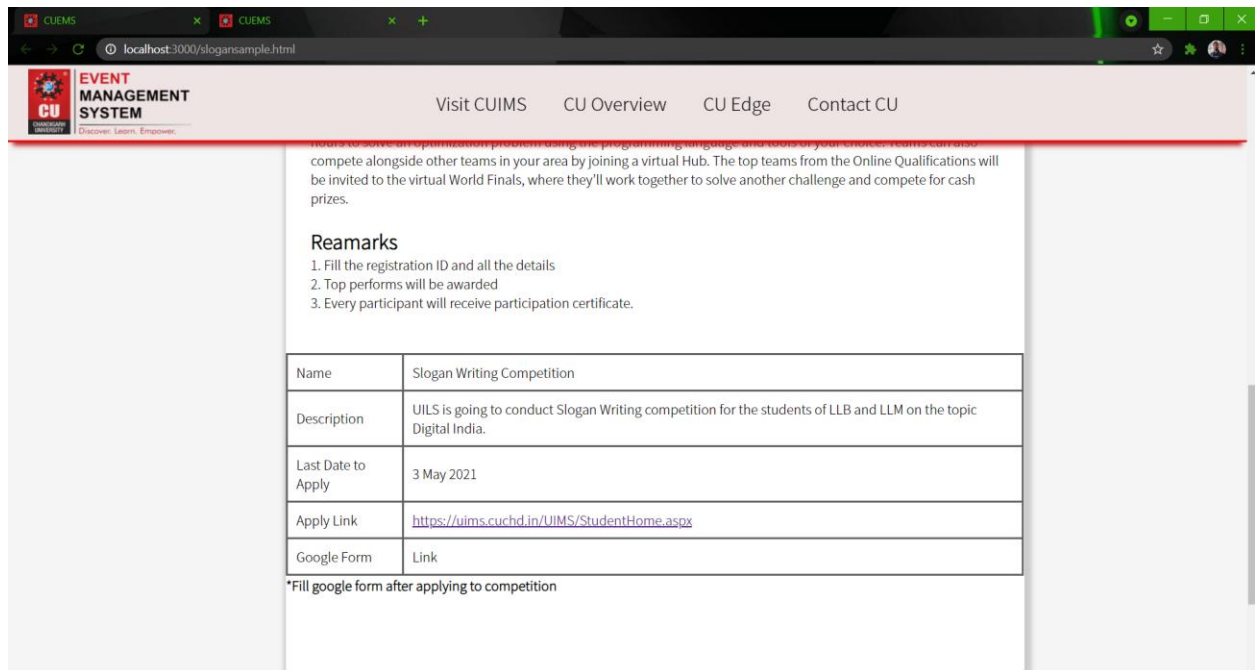
**Figure 15 : CU-EMS sample technical event page (3/3)**

## 1.5.2. Non-Technical Events :

Below is the sample page of a Non-Technical Event.



**Figure 16 : CU-EMS sample non-technical event page (1/2)**

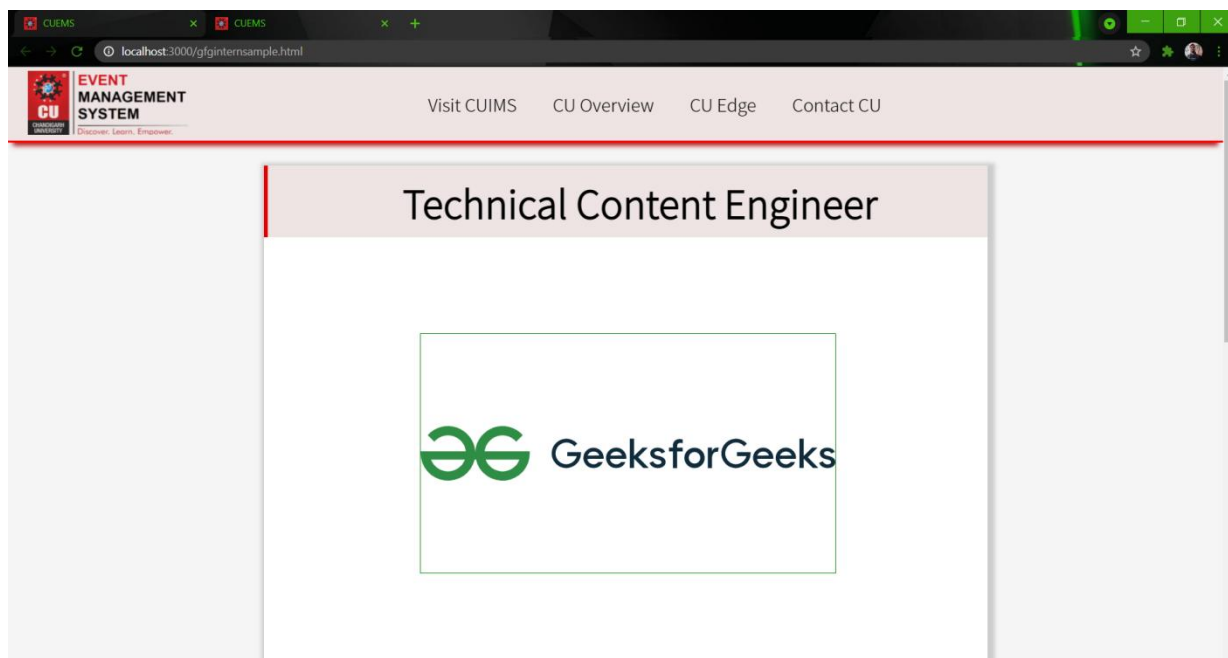


**Figure 17 : CU-EMS sample non-technical event page (2/2)**

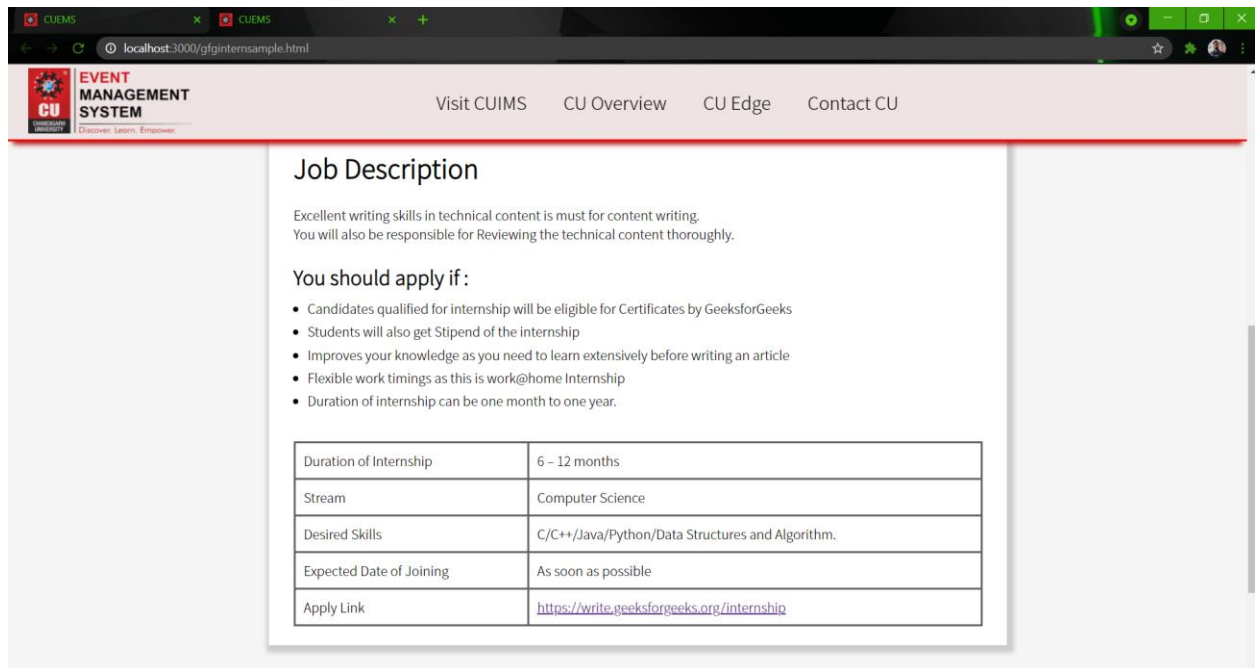
## 1.6. Others :

### 1.6.1. Internships :

Below is the sample page of Internship.



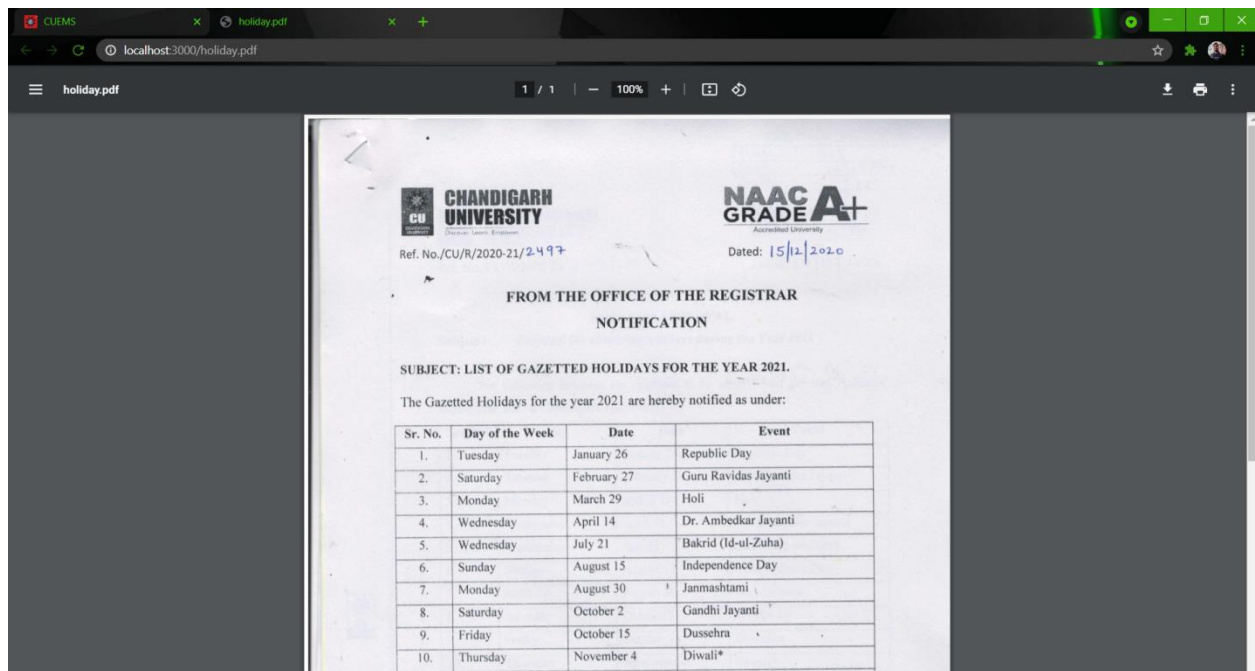
**Figure 18 : CU-EMS sample internship page (1/2)**



**Figure 19 : CU-EMS sample internship page (2/2)**

## 1.6.2. Notices :

Below is the sample of a notice. It redirects to another page which contains the direct pdf file of notice.

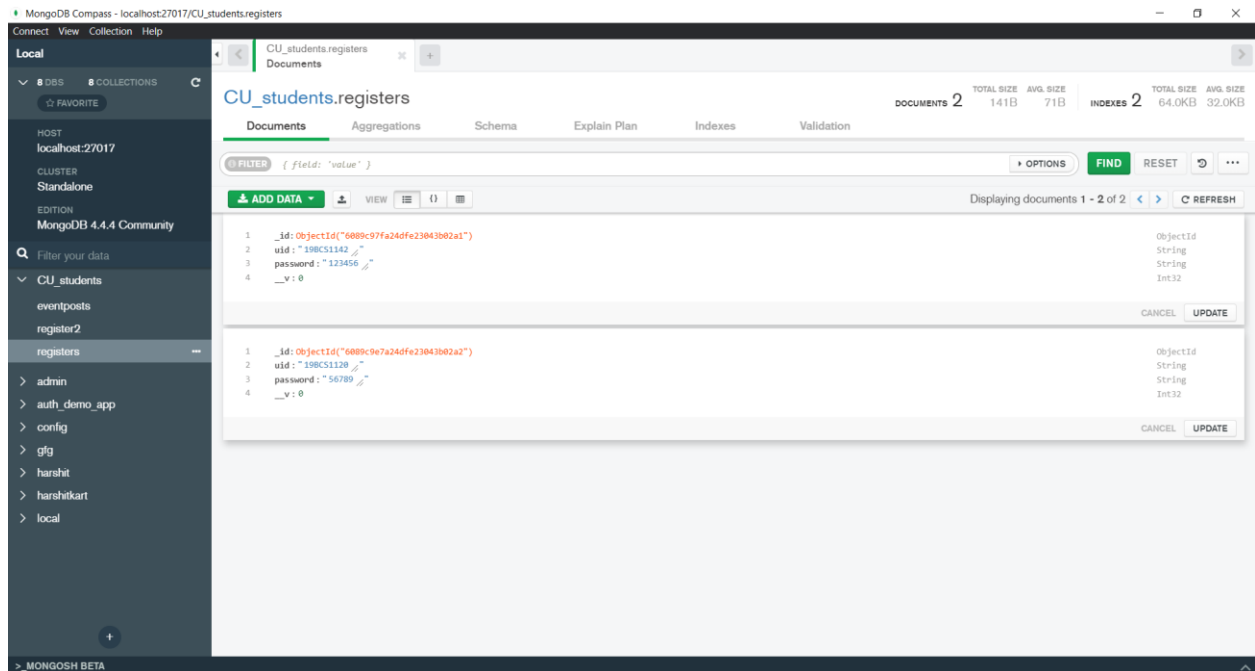


**Figure 20 : CU-EMS sample notice redirecting page**

## 2. Output validation and comparison

### 2.1. Student Login page validation with backend

#### 2.1.1. Student's sample database of login credentials



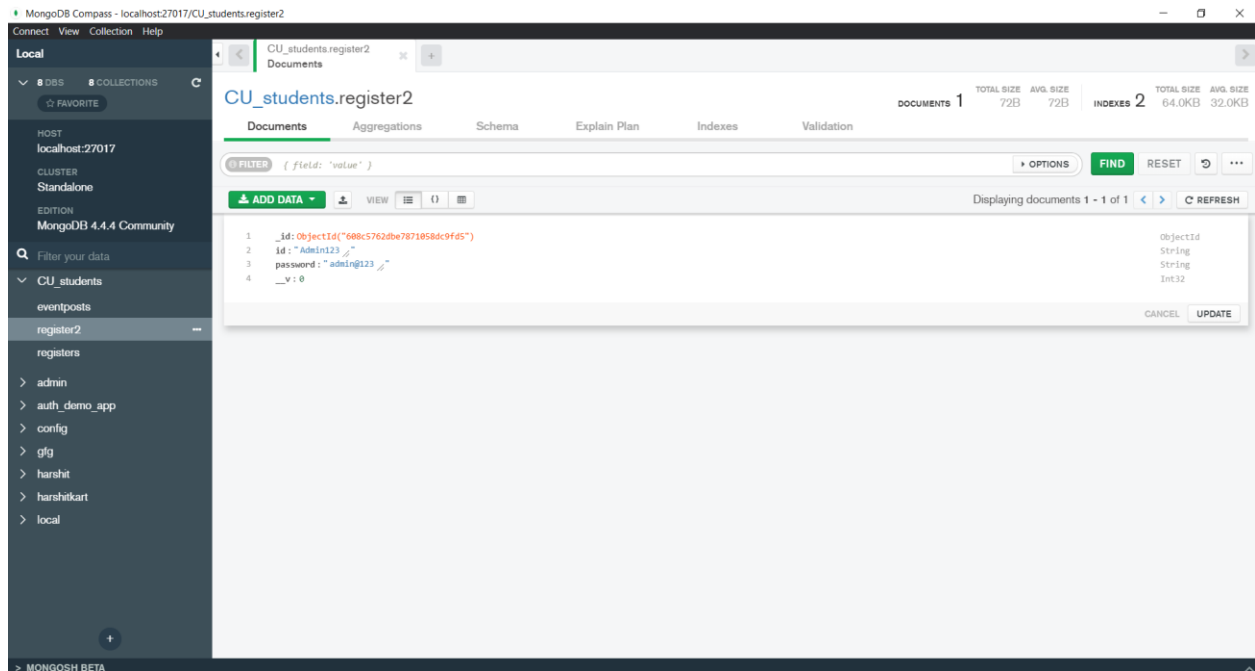
**Figure 21 : CU-EMS Student Database**

#### 2.1.2. Validation/Verification of the above login credentials

```
app.post("/login", async (req,res) => {
  try {
    const uid = req.body.uid;
    const password = req.body.password;
    const studentuid = await Register.findOne({uid:uid});
    if(studentuid.password === password){
      res.status(201).render("index2");
    }else{
      res.send("Invalid password");
    }
  } catch (error) {
    res.status(400).send("Invalid UID");
  }
});
```

## 2.2. Admin Login page validation with backend :

### 2.2.1. Admin's sample database of login credentials :



**Figure 22 : CU-EMS Admin Database**

### 2.2.2. Validation/Verification of the above login credentials :

```
app.post("/login2", async (req,res) => {
  try {
    const id = req.body.id;
    const password = req.body.password;
    const adminid = await Register2.findOne({id:id});
    if(adminid.password === password){
      res.status(201).render("index3");
    }else{
      res.send("Invalid password");
    }
  } catch (error) {
    res.status(400).send("Invalid ID");
  }
});
```



## 2.3. Validation with backend for posting a new Event :

### 2.3.1. Saving the event posted by admin in the database :

**Input :**

*eventpost.js*

```
const mongoose = require("mongoose");
const studentSchema = new mongoose.Schema({

  myName : {
    type:String,
    required:true
  },
  myEvent : {
    type:String,
    required:true
  },
  eventType :{
    type:String,
    required:true
  },
  myRemark :{
    type:String,
    required:true
  },
  myDate :{
    type:String,
    required:true
  },
  appt :{
    type:String,
    required:true
  },
  myDate1 :{
    type:String,
    required:true
  },
  appt1 :{
    type:String,
    required:true
  },
  myLink :{
    type:String,
    required:true
  },
  myLink1 :{
    type:String,
```

```

        required:true
    },
    img :{
        type:String,
        required:true
    }
})

const eventpost = new mongoose.model("eventpost", studentSchema);
module.exports=eventpost;

```

The below line must(compulsory) in the html code of the event form :

```

<form action="/eventpost" method="POST">

```

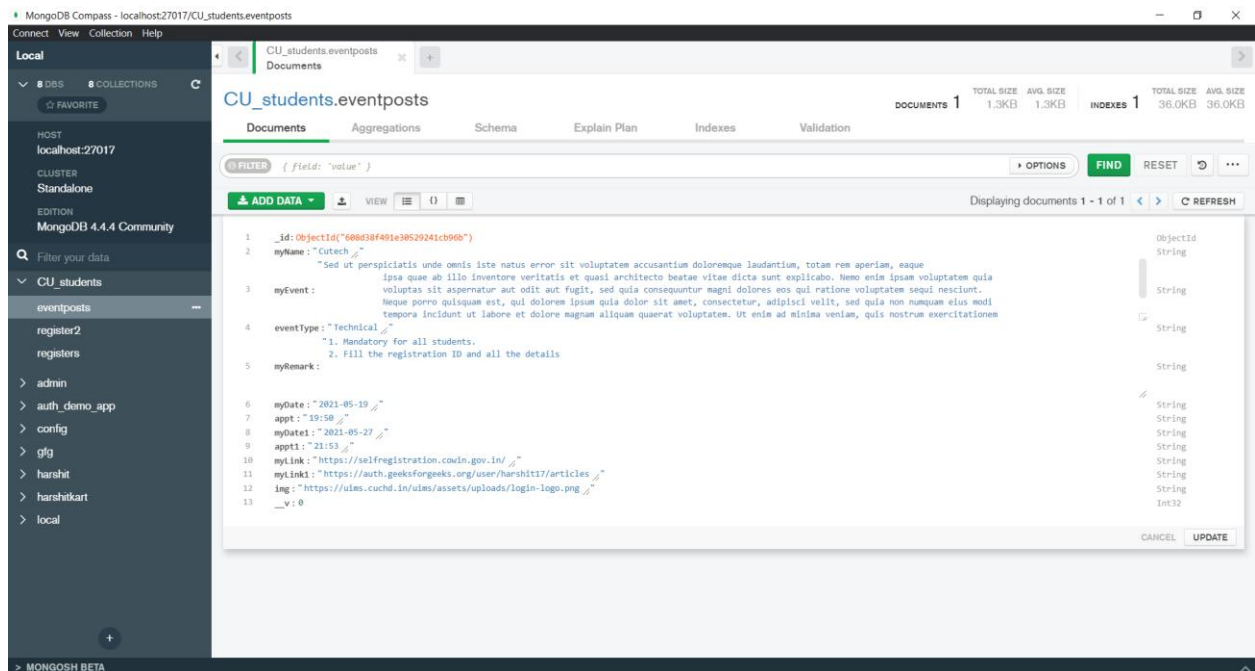
### 2.3.2. Validation of the event form

```

app.post("/eventpost", async (req,res) => {
    try {
        const registerevent = new eventpost({
            myName : req.body.myName,
            myEvent : req.body.myEvent,
            eventType :req.body.eventType,
            myRemark :req.body.myRemark,
            myDate :req.body.myDate,
            appt :req.body.appt,
            myDate1 :req.body.myDate1,
            appt1 :req.body.appt1,
            myLink :req.body.myLink,
            myLink1 :req.body.myLink1,
            img :req.body.img
        })
        const registeredevent = await registerevent.save();
        res.status(201).render("index3");
    } catch (error) {
        res.status(400).send(error);
    }
});

```

### 2.3.3. Verification of the event form :



**Figure 23 : CU-EMS Database of upcoming events made by admin**

The event gets saved in the database and after the event's details are verified, we will be posting in the site.

## **Team Work :**

<b>Name</b>	<b>UID</b>	<b>Class/Sec</b>	<b>Tasks Done</b>
Harshit Srivastava	19BCS1142	CSE 2A	<ul style="list-style-type: none"><li>• Planning, designing and implementation of validation and verification part of the project</li><li>• Front End part using HTML, CSS, JS and Bootstrap.</li><li>• Backend part of the website using Nodejs, Nodemon, Express, Mongoose and MongoDB</li><li>• Documentation</li></ul>
Tanmay Tripathi	19BCS1120	CSE 1C	<ul style="list-style-type: none"><li>• Designing and development of the front end pages using HTML, CSS, and JS</li><li>• Worked on things like tags, keywords, logo and timers using complex JS libraries.</li><li>• Documentation</li></ul>