University Forum Website Software Requirements Specification

Based on IEEE 830-1998 Standards

Version: 1.0
Date: December 14, 2025
Status: Approved for Development

Prof:
Wessam Ahmed

Students:

Eman Gaber
Renada Ahmed
Mariam Ahmed
Mostafa Mahmoud
Mohamed Abu-Alhajjaj
Ahmed Saad

1. Introduction

1.1 Purpose

This document specifies the requirements for the University Forum Website. The purpose is to:

· Define what the system will do for developers, managers, and users
· Provide a basis for software development and testing
· Ensure clear understanding of the final product

1.2 Product Scope

The University Forum Website is a complete web application that connects students with university administration, providing:

· Students: Event registration, lost item reporting, club communication
· Administration: Event management, report approval, statistics tracking

What the system includes:

· Secure login system
· Event management (create, edit, delete)
· Lost & Found system
· About Us and Contact Us pages

What the system does NOT include:

· Payment system
· Live chat
· Course management
· Works only as a website (no mobile app)

1.3 Glossary

Term	Definition
Student	A regular user with limited permissions. Students can view upcoming and past events, view Lost & Found items, and access general information pages. Students *cannot* create Lost & Found reports.
Admin	The system administrator with full access. Admins can manage events (create/edit/delete), upload past event photos, and fully manage Lost & Found items.
Event	A university club activity. Events can be upcoming (future activities students can join) or past (completed activities with descriptions and photos).
Upcoming Events	Events planned for the future. Students can view and join these events.
Past Events	Events that have already happened and include descriptions and photos.
Lost & Found	A module in the system managed **only by the admin**. Admins can add, edit, or delete lost or found items. Students can only *view*.
Join Event	An action performed by a student to register for an upcoming event.
JWT (JSON Web Token)	A secure authentication token used to verify user identity after login.
API	Backend endpoints that the frontend communicates with to send or retrieve data.
Frontend	The user-facing part of the system, built using React.
Backend	The server-side logic built using Express.js and connected to PostgreSQL through Prisma.
Prisma ORM	A database toolkit used for managing and accessing the PostgreSQL database.
PostgreSQL	The database system used to store all system data.
About Us Page	A page displaying club overview, mission, vision, team members, and achievements timeline.
Contact Us Page	A page containing communication methods such as Facebook, WhatsApp, Email, and phone number.

1.4 References

1. React Official Documentation
2. Express.js Documentation
3. PostgreSql Documentation

1.5 Overview

This document contains:

· Section 1: Project introduction
· Section 2: General system description
· Section 3: Detailed requirements
· Section 4: Requirements analysis

2. General Description

2.1 Product Perspective

The product will be a standalone website that runs on:

· Browsers: Chrome
· Devices: Computer, laptop
· Servers: Any server supporting Node.js

Important Note: This is a website only, requiring internet browser access. It works on laptops,and computers browsers.

2.2 Product Functions

1. Registration and Login
   · New student registration
   · User login
   · Password protection
2. Event Management
   · Display upcoming events
   · Display past events
   · Student event registration
3. Lost & Found System
   · Report lost items
   · Report found items
   · Claim items
4. Admin Dashboard
   · Add new events
   · Approve reports

2.3 User Characteristics

Student:

· Age: 18-25 years
· Technical experience: Medium (internet use)
· Needs: Easy to use, fast performance
· Goals: Join events, find lost items

Admin:
· Age: 25-60 years
· Technical experience: Advanced
· Needs: Advanced management tools, accurate reports
· Goals: Organize events, track reports

2.4 Constraints

1. Technical Constraints:
   · Must work on modern browsers
   · Run on PC or Laptops
   · Must support English
   · Requires laptop/internet connection to work
2. Time Constraints:
   · Development time: 1 months
   · Testing time: 1 weeks
3. Financial Constraints:
   · Use free technologies
   · Minimum operating costs

2.5 Assumptions and Dependencies

Assumptions:

1. University provides good internet connection
2. Users have modern devices
3. University system doesn't change during development
4. Users have laptop or computer access

Dependencies:

1. Image hosting service (Cloudinary)
2. Database server
3. Domain name for website
4. Node.js installed on server

---

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

Registration Page:

· Fields: Name, email, password, confirm password
· Button: "Create Account"
· Links: "Already have an account? Login"

Login Page:

· Fields: Email, password
· Button: "Login"

Home Page:

· Upcoming events section
· Past events section
· Navigation bar
· Footer
. Lost and found section

3.1.2 Software Interfaces

Operating System: Windows 10/11
Browsers: Chrome 90+
Database: postgreSQL
Programming Language: JavaScript (express)

3.2 Functional Requirements

Registration and Login:

1. Register new student
   · Enter personal data
   · Verify email validity
   · Confirm password
   · Save data to database
2. User login
   · Verify email and password
   · Create secure login token (JWT)
   · Redirect user to home page

Event Management:

1. Display events
   · Show upcoming events
   · Show past events
   · Filter events by date
2. Student joins event
   · Add student to list
   · Send confirmation to student
3. Add new event (Admin only)
   · Data entry form
   · Image upload
   · Set event date

Lost & Found System:
1. admin add lost and found report
  . Select report type (Lost / Found)
  . Enter item description
  . Upload item photos
  . Set location and date
  . Publish the report to be visible to users

2. User views lost and found items
   · view all lost and found item
   · Browse available rebort
3.user contact admin to claim item
   · select the item the user believe belongs to them
   · send a contact or claim request to the admin
   · wait for the admin response without further action
3. Admin manage claim
   . Receive user contact/claim request
   . communicate with the user to verify the ownership
   . Approve or reject the claim
   . Send notification to the user with the final decision

3.2.1 Scenario: Student Joins an Event

User: Student
Before Start: Student is logged in, event is upcoming
Steps:

1. Student goes to home page
2. Sees upcoming events section
3. Clicks on desired event
4. Clicks "Join Event" button
5. System confirms registration

After Completion: Student is registered for event

3.2.2 Scenario: Admin Creates Event

User: Admin
Before Start: Admin is logged in
Steps:
1. Admin goes to dashboard
2. Clicks "Event Management"
3. Clicks "Add New Event"
4. Enters event data (title, description, date, location)
5. Uploads event image
6. Clicks "Save"
7. System saves event

After Completion: Event appears for students

3.2.3 Scenario: Student search Lost Item

User: Student
Before Start: Student is logged in
Steps:

1. Student goes to the lost and found page page
2. Student views the list of found item
3. Student selects an item that matches their lost item5. Uploads photos
4. Student clicks contact admin
5.Student sends a message to the admin with ownership details
6.System sends the request to the admin

After Completion: The admin receives the student’s request and contacts the student

3.2.4 Scenario: Admin Manages Lost & Found

User: Admin
Before Start: Admin is logged in
Steps:

1.Admin opens lost found management.
2. Admin clicks add new report
3. Admin enters item details (title, description, location, date).
4. Admin uploads item photos
5. Admin clicks publish report
6. System saves and publishes the report
7. Clicks "Reject" for incorrect reports

After Completion: The found item is visible to all students in the Lost & Found page

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

Function	Maximum Time
Home page load	3 seconds
Login process	2 seconds
Image upload	5 seconds
Data search	1 second

3.3.2 Security Requirements

1. Data protection
   · Use HTTPS
   · Protection from hacking
2. Access control
   · Different user permissions
   · Identity verification for each operation
   · Login activity logging
3. Privacy protection
   · Don't share personal data
   · Delete old data
   · Account deletion option

3.3.3 Usability Requirements

1. Simple design
   · Clear and easy interface
   · Appropriate colors
   · Readable fonts
2. Mobile responsive
   · Good display on phones
   · Big buttons
   · Proportional images
3. Technical support
   · Clear error messages
   · User guide
   · Technical support

3.3.4 Documentation Requirements

1. User guide
   · How to register
   · How to add events
   · How to report items
2. Developer guide
   · Code explanation
   · Database explanation
   · Installation method

3.4 Deleted Requirements

Requirements that were canceled:

1. Live chat system
   · Reason: Needs long development time
   · Alternative: Use email
2. Rating system
   · Reason: Not necessary in first phase
   · Alternative: Add in later updates

4. Requirements Analysis

4.1 Analysis Team and Roles

Member Role Responsibilities
Project Manager Planning Follow schedule
Systems Analyst Requirements gathering Understand user needs
Interface Designer Design final appearance
Frontend Programmer Frontend development Program user interface
Backend Programmer Backend development Program server
Quality Tester Testing Check software before delivery

4.2 Analysis Process

1. User interviews
   · Interview with students
   · Interview with administration
   · Collect notes
2. Study similar systems
   · Other university websites
   · Student forums
   · Identify advantages and disadvantages
3. Determine priorities
   · Important and urgent first
   · Important but not urgent second
   · Not important but urgent third
   · Not important and not urgent last

4.3 Dependency Analysis

4.3.1 Business Requirements Analysis

Business Need Technical Requirement Priority
Student participation Event system High
Find lost items Report system High
Club introduction About Us page Medium
Communication Contact Us page Medium

4.3.2 Risk Analysis

Requirement Risks Treatment Method
Image upload Large size Set maximum size
Registration Fake data Email verification
Login Account hacking Strong passwords
Saving Data loss Backup

4.4 Problems Raised

1. Problem: Some students don't have university email
   Solution: Allow any email address
2. Problem: Difficulty for beginners
   Solution: Simple design and user guide

4.5 Verification of Original Requirements

Original Requirement Status Notes
Student registration ✓ Confirmed Basic
Event management ✓ Confirmed Basic
Lost & Found system ✓ Confirmed Basic
Admin dashboard ✓ Confirmed Basic
Responsive design ✓ Confirmed Necessary
Social media links ✓ Confirmed Required
Team section ✓ Confirmed For introduction
Contact information ✓ Confirmed Basic

Appendix A: Technical Details

A.1 Programming Interface (API)

Registration and Login:

```
POST   /api/signup      # Create new account
POST   /api/login       # Login
POST   /api/logout      # Logout
```

Events:

```
GET    /api/events      # Get events
POST   /api/events      # Add event (Admin only)
PUT    /api/events/:id  # Edit event (Admin only)
DELETE /api/events/:id  # Delete event (Admin only)
POST   /api/events/:id/join  # Join event
```

Lost Items:

```
GET    /api/lost-items      # Get reports
POST   /api/lost-items      # Add new report
PUT    /api/lost-items/:id  # Edit report
POST   /api/lost-items/:id/claim  # Claim item
```

A.2 Database Tables

Users Table:

javascript
```
{
  _id: ID number,
  email: Email address,
  password: Password (encrypted),
  role: Role (student/admin),
  name: Name,
  phone: Phone,
  studentId: University ID
}
```

Events Table:

javascript
```
{
  _id: ID number,
  title: Event title,
  description: Description,
  date: Date,
  location: Location,
  image: Event image,
  type: Type (upcoming/past)
}
```

Reports Table:

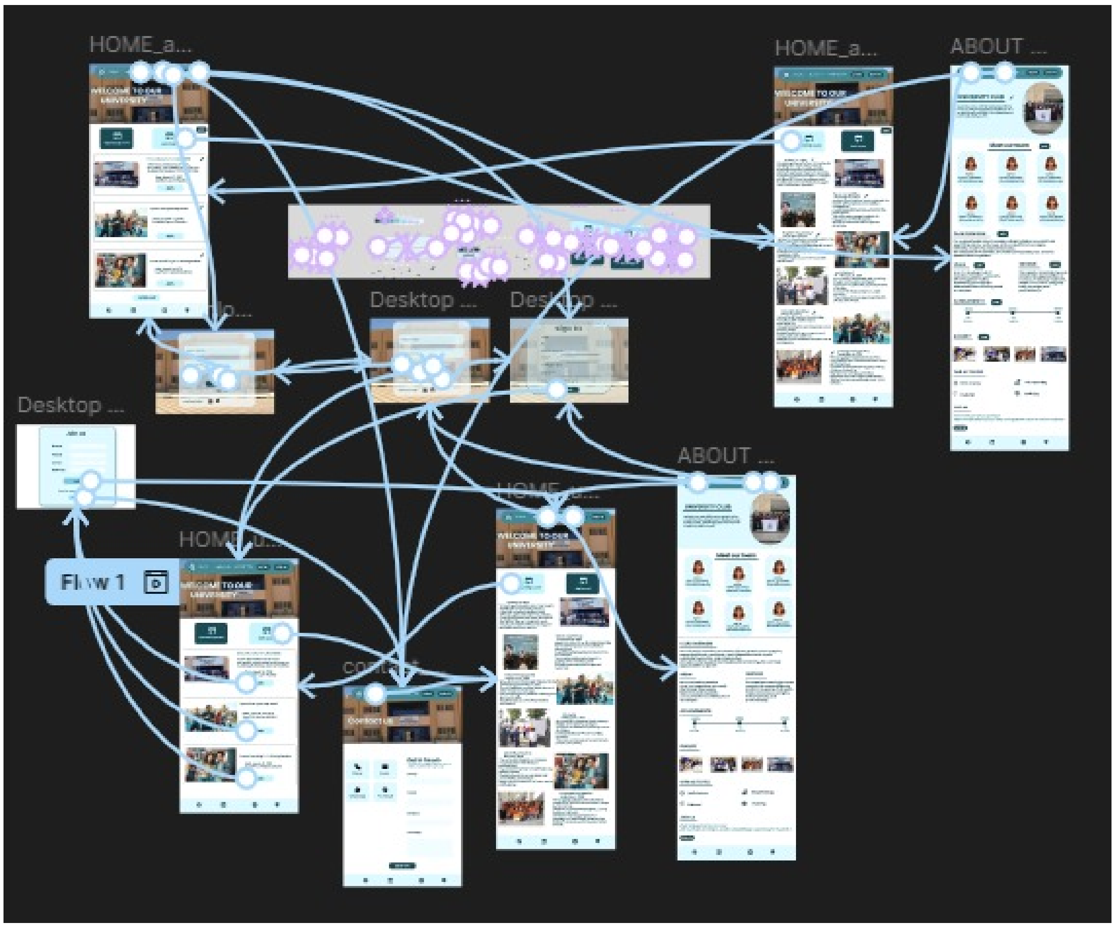
javascript
```
{
  _id: ID number,
  title: Report title,
  description: Description,
  type: Type (lost/found),
  location: Location,
  date: Date,
  images: [Photos],
  status: Status (new/approved/claimed),
  reportedBy: Reporter,
  claimedBy: Claimant,
  createdAt: Creation date
}
```

A.3 Permission Map

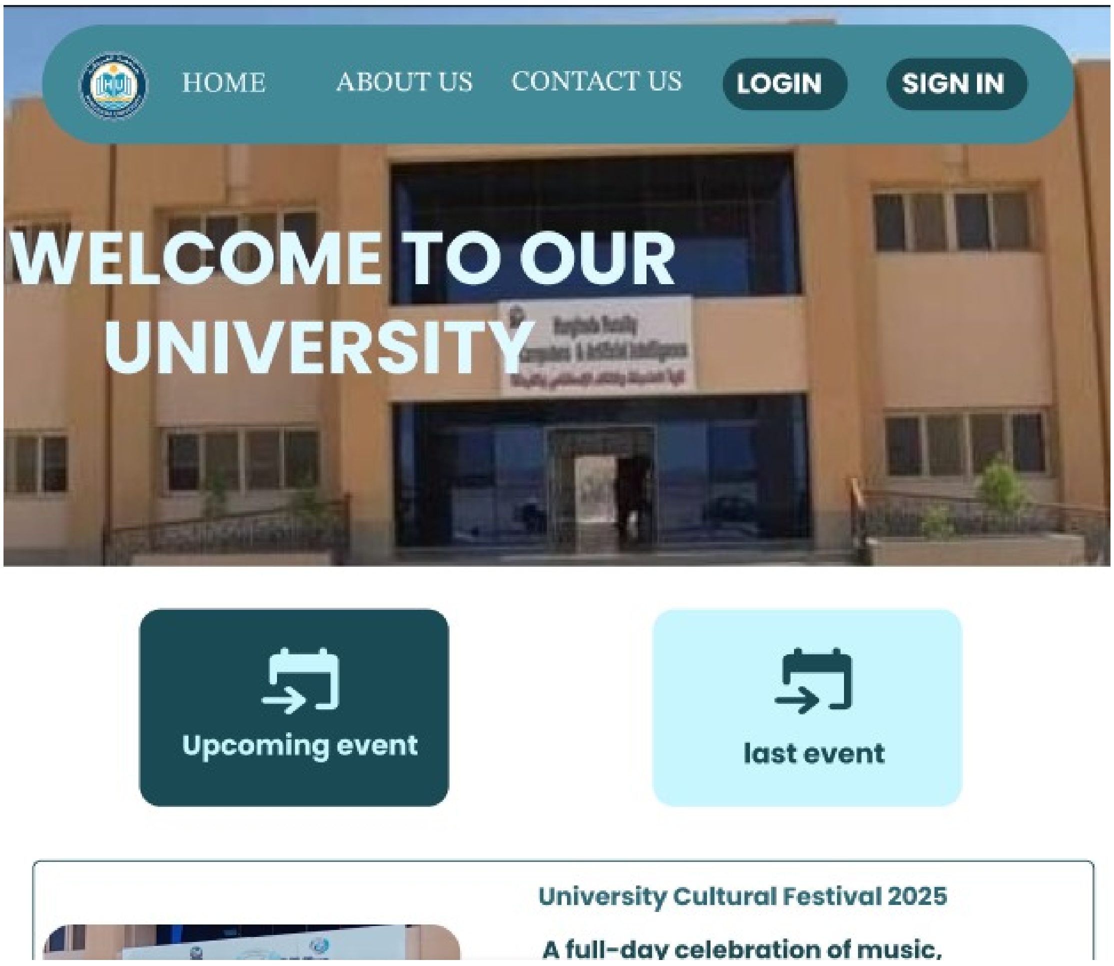
Action	Student	Admin	Visitor
View home page	✓	✓	✓
View events	✓	✓	✓
Join event	✓	✗	✗
Report lost item	✓	✗	✗
Edit profile	✓	✓	✗
Add event	✗	✓	✗
Delete event	✗	✓	✗
Approve reports	✗	✓	✗
View dashboard	✗	✓	✗
Manage users	✗	✓	✗

Appendix B: Interface Design

Figma


HOME_a...
HOME_a...
ABOUT ...
Desktop ...
Desktop ...
Desktop ...
ABOUT ...
HOME_a...
HOME_a...
Flow 1
contact
WELCOME TO OUR UNIVERSITY
Contact us

Home Page

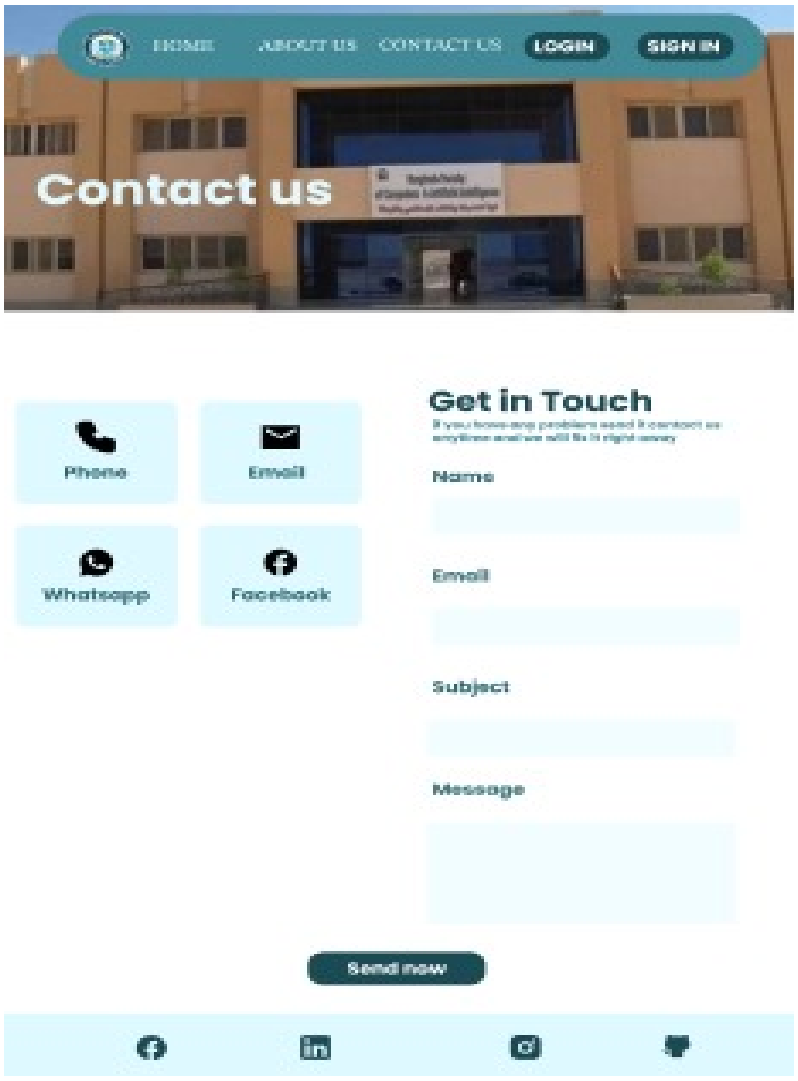


WELCOME TO OUR UNIVERSITY

University Cultural Festival 2025

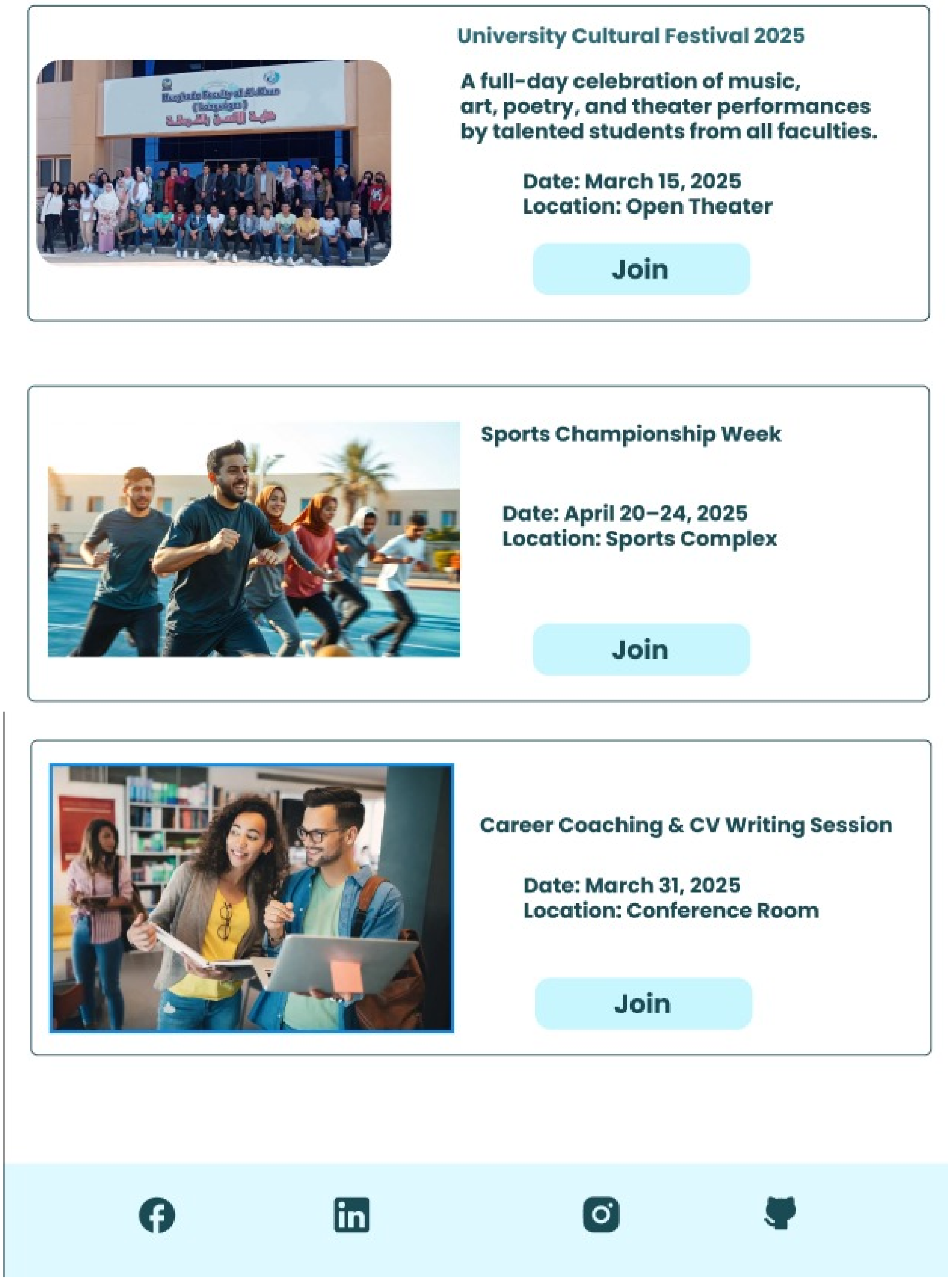
A full-day celebration of music,

Contact Us



If you have any problem send it contact us anytime and we will fix it right away

Upcoming Events



University Cultural Festival 2025

A full-day celebration of music, art, poetry, and theater performances by talented students from all faculties.

Date: March 15, 2025
Location: Open Theater

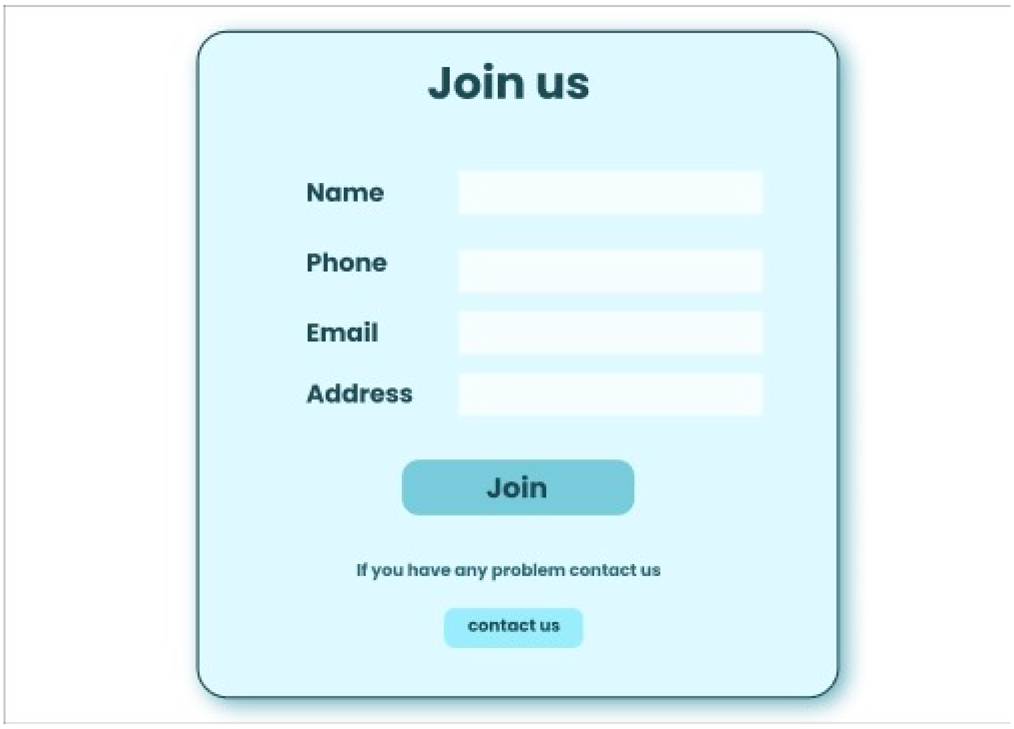
Sports Championship Week

Date: April 20–24, 2025
Location: Sports Complex

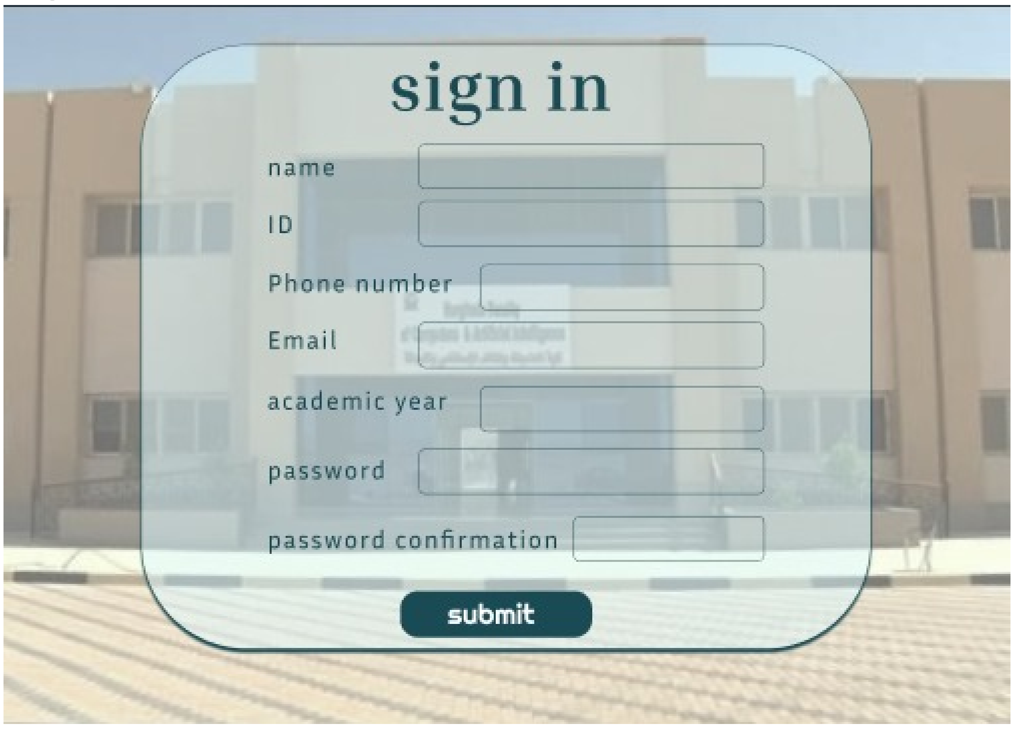
Career Coaching & CV Writing Session

Date: March 31, 2025
Location: Conference Room

Join to Upcoming events



Sign in the site



sign in

name

ID

Phone number

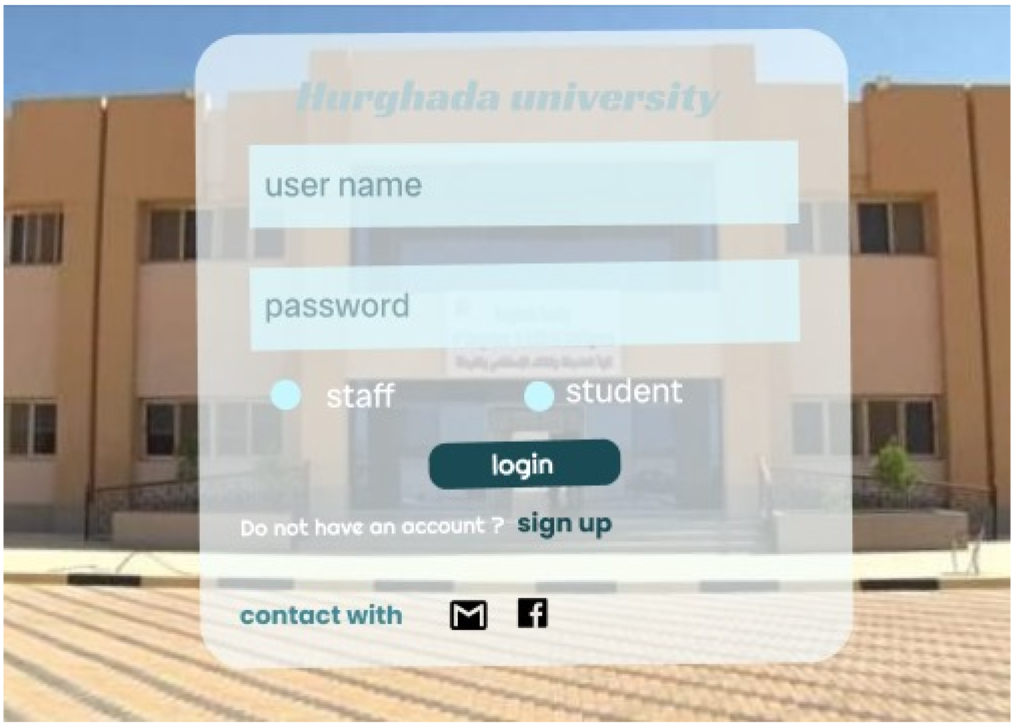
Email

academic year

password

password confirmation

Login


Hurghada university
user name
password
staff
student
login
Do not have an account ? sign up
contact with

Installation Requirements

For Users:

1. Device: Laptop or computer (Windows, macOS, or Linux)
2. Browser: Chrome
3. Internet: Stable connection
4. No installation needed: Just open website in browser

For Developers:

1. Node.js
2. MongoDB
3. Code Editor
4. Git
5. Terminal/Command Prompt

For Server:

1. Node.js: Version
2. PostgreSQL: Installed and running
3. Code Editor
4. Git
5. Terminal/Command Prompt
6.Postman

Important: The system works as a website only. Users access it through their browser. No software installation is needed on user devices.