## **Software Requirements Specification (SRS)**

**Project Title:** LGU Event Management Portal  
 **Project Type:** Full Stack Web Application

### **1. Introduction**

**Purpose:** To develop a full-stack web application that enables Lahore Garrison University (LGU) to manage events like workshops, seminars, and student society activities efficiently.

**Intended Users:**

* University students
* Event organizers
* Admin users (optional authentication)

### **2. Functional Requirements**

#### **2.1 Event Listing (Frontend + Backend)**

* Display upcoming events with:  
  + Title
  + Date
  + Department
  + Description
* Filter events by department (e.g., SE, CS, IT)

#### **2.2 Admin Panel (Optional Authentication)**

* Add, edit, delete events using a form-based interface.

#### **2.3 Student Registration for Events**

* Allow student registration with:  
  + Name
  + Roll number
  + Email
* Admin-only view of registered students.

### **3. Non-Functional Requirements**

* **Usability:** Clean and intuitive UI for both users and admin.
* **Performance:** Fast loading times and smooth event filtering.
* **Security:** Basic authentication for admin features (optional).
* **Portability:** Should be deployable on a web server or hosted on GitHub.

### **4. Bonus (Must Have) Features**

* Admin authentication (session/token-based)
* Confirmation email (mock or real)
* Search bar for events
* Export event registrations as CSV

## **🕒 Time Budget: 2 Hours (120 minutes)**

### **🔹 Phase 1: Setup & Database Design (10 minutes)**

**Objective:** Create project folders and MySQL database with necessary tables.

#### **✅ Steps:**

* Set up the Laravel/PHP/Express project (laravel new event-portal or npm init).
* Create MySQL DB lgu\_event\_portal.
* Create tables:  
  + events: id, title, date, department, description
  + students: id, name, roll\_number, email
  + registrations: id, student\_id, event\_id
  + admins: id, email, password

### **🔹 Phase 2: Backend - Event CRUD API (20 minutes)**

**Objective:** Build event listing and admin CRUD.

#### **✅ Steps:**

* Create RESTful APIs:  
  + GET /events – list events
  + POST /events – add event
  + PUT /events/{id} – update event
  + DELETE /events/{id} – delete event
* Add filter by department query (?department=CS).
* Use Laravel Eloquent or raw SQL queries.

### **🔹 Phase 3: Frontend - Event Listing UI (15 minutes)**

**Objective:** Show events and implement filtering by department.

#### **✅ Steps:**

* Basic Bootstrap UI to show:  
  + Title, Date, Dept, Description
* Add dropdown for department filter (onchange => fetch filtered events).
* Use AJAX/Fetch API to retrieve events from backend.

### **🔹 Phase 4: Admin Panel (Add/Edit/Delete Events) (15 minutes)**

**Objective:** Implement form-based admin UI.

#### **✅ Steps:**

* Create login page for admin (if using sessions/tokens).
* Admin dashboard:  
  + Form to add event
  + List to edit/delete events
* Store token/session upon login to protect routes (optional but ideal).

### **🔹 Phase 5: Student Registration Module (15 minutes)**

**Objective:** Event registration form + view registered users.

#### **✅ Steps:**

* Create registration form with:  
  + Name, Roll No., Email
* On submit, POST to /register with event\_id.
* Create a page (admin only) to list registered students for each event.

### **🔹 Phase 6: Bonus Features Implementation (30 minutes)**

**Objective:** Cover all optional features for full marks.

#### **✅ Steps:**

* ✅ **Admin Authentication**
  + Simple login with session or token
* ✅ **Confirmation Email**
  + Use mock (just show alert or log to console/email sent)
* ✅ **Search Bar for Events**
  + Add input box + filter results client-side or via /search?q=seminar
* ✅ **Export Registrations as CSV**
  + Create button for admin to download CSV of event registrations

### **🔹 Phase 7: Final Touches & Deployment (15 minutes)**

**Objective:** Polish UI, test, and prepare submission.

#### **✅ Steps:**

* Final CSS polish (make Bootstrap cards/tables look good).
* Check all links & forms work.
* Push to GitHub and copy demo link.
* Prepare short README:
  + Project setup
  + How to test each feature

LGU-EVENT-PORTAL/

│

├── 📁 assets/ # CSS, JS, images

│ ├── css/

│ │ └── style.css

│ ├── js/

│ │ └── scripts.js

│ └── img/

│ └── logo.png (optional)

│

├── 📁 includes/ # Reusable files (e.g., DB connection, header, footer)

│ ├── db.php # DB connection script

│ ├── header.php # Common header (e.g., navbar)

│ └── footer.php # Common footer

│

├── 📁 admin/ # Admin panel: event CRUD, auth

│ ├── login.php

│ ├── logout.php

│ ├── dashboard.php

│ ├── add\_event.php

│ ├── edit\_event.php

│ ├── delete\_event.php

│ ├── registrations.php # View registered students

│ └── export\_csv.php # Export registrations to CSV

│

├── 📁 events/ # Public event features

│ ├── index.php # List events + filter + search

│ └── register.php # Student registration form

│

├── 📁 functions/ # Utility functions

│ ├── auth.php # Admin session handling

│ ├── email.php # Mock email sender

│ └── helpers.php # Reusable functions

│

├── 📁 sql/ # Database setup

│ └── lgu\_event\_portal.sql # SQL schema export

│

├── .htaccess # (Optional) URL rewriting

├── README.md # Project overview and setup instructions

└── index.php # Entry point – redirect to /events/index.php