# EEVO

**Event & Election Virtual Organizer (Web App)**

**Tagline**: One platform for department events, cross-department sports & culturals, elections, finances, and notifications — following real college rules.



**Mentor: Mrs.SOWMIYA AP/CSE**

**TEAM MEMBERS**

**Mr.TAMIZHMANI V DME…,**

**Mr.KANISH KUMAR P**

## Abstract

EEVO Web is a responsive, role-based platform built with Java + Spring Boot and a lightweight frontend stack (HTML, CSS, JavaScript). It enables students and staff to manage events, elections, and finances with full compliance to real college protocols.  
Now enhanced with **collaborative event creation**, EEVO Web supports multi-departmental coordination and shared authority across roles.

## Project Description

EEVO Web provides a secure, scalable interface for managing college operations. It supports departmental workshops, inter-departmental events, and student elections, modeled on real college protocols.  
Users are assigned roles—HOD, PT Sir, Event Coordinator, Election Commissioner, Treasurer, or Student—with tailored permissions. This enables decentralized management with centralized oversight.

### Core Features:

* **Event Management**: Create, approve, and monitor events, including collaborative events across departments or individuals.
* **Election Workflow**: Nomination to inauguration with secure voting and result analysis.
* **Finance Tracking**: Upload and verify bills linked to events/elections.
* **Push Notifications**: Real-time updates via web alerts or email.
* **Audit Logging**: Track actions for compliance and transparency.

## Collaborative Event Feature

## Feature Overview

EEVO Web supports **inter-departmental and individual collaborations** during event creation. Organizers can select one or more departments or individuals to co-host an event. This ensures shared visibility, joint responsibility, and synchronized authority across all collaborators.

### Functional Highlights

* **Collaboration Toggle**: Enable “Collaborate with…” during event creation.
* **Dual Visibility**: Event appears in dashboards of all selected departments and their Presidents.
* **Shared Authority**:
  + Both Presidents can approve, modify, or cancel the event.
  + Finance uploads and notifications are visible to all collaborators.
* **Audit Trail**: Collaboration metadata is logged for transparency and future audits.

## Key Roles & Authorities

| **Role** | **Responsibilities** |
| --- | --- |
| HOD | Schedule elections, preload staff IDs, manage staff registry, approve configurations |
| Department President | Create/modify departmental events, authorize organizers, approve collaborative events |
| Association Member | Create/modify events when permitted |
| Election Commissioner | Manage election lifecycle, assign post-election roles |
| PT Sir | Create/manage sports events, assign Sports Leaders |
| Sports Leader | Manage sport-specific activities |
| Event Coordinator | Manage cultural events (appointed by Principal) |
| Treasurer | Upload bills/receipts, manage finance |
| Staff/Student Viewer | View-only access; voting rights as per rules |
| Event Organizer (Staff) | Create/mentor career guidance/workshops |

## Requirements Analysis

### Functional Requirements:

* Role-based event creation and management
* Election eligibility and scheduling
* Push/email notifications for events and election stages
* Finance uploads and payment tracking
* Year advancement and audit logging
* Collaborative event creation with shared visibility and authority

### Non-Functional Requirements:

* Secure authentication and role-based authorization
* Responsive UI for desktop and mobile browsers
* CI/CD integration for web builds
* API integration with backend services
* SEO optimization and accessibility compliance

## Election Workflow

1. **Nomination**: 3-day window for 4th-year students
2. **Campaigning**: 5 days with enforced protocols
3. **Analysis**: 1 day for validation
4. **Voting**: 24-hour secure ballot
5. **Result**: Auto-analysis + manual role assignment
6. **Inauguration**: Scheduled by Commissioner
7. **Year Rollover**: Admin-confirmed or auto-incremented

## Design Highlights

* Role-based navigation with dynamic menus
* Secure voting interface with real-time status
* Push/email notification integration
* Finance upload with image and PDF support
* Configurable election rules and timers
* Collaborative event creation UI with multi-select department/individual picker
* Event cards display collaboration badge and shared approval status

## Detailed Implementation

### Web Frontend

* **Tech**: HTML, CSS, JavaScript (Vanilla or Bootstrap)
* **Structure**:
  + Modular pages for each role (HOD, PT Sir, Commissioner, etc.)
  + Reusable components: forms, tables, modals
  + Responsive design for desktop and mobile
* **Routing**: Spring Boot MVC or client-side routing with hash-based navigation
* **Validation**: JavaScript-based form validation with role-specific logic
* **Notifications**: Web push or email via backend integration
* **Collaborative Events**:
  + Multi-select dropdown for departments/individuals
  + Collaboration badge in event listings
  + Shared approval logic in UI

### Backend

* **Tech**: Java + Spring Boot
* **Endpoints**:
  + /api/web/events
  + /api/web/elections
  + /api/web/notifications
* **Authentication**: Spring Security + JWT
* **Collaborative Events**:
  + Extend event schema with collaborators[]
  + Normalize with junction table: event\_collaborators(event\_id, dept\_id, user\_id)
  + Update fetch logic to include all relevant departments

### API Docs

* **Tech**: Swagger / SpringDoc OpenAPI
* **Integration**: Auto-generated docs for frontend consumption and testing

### Authentication

* **Tech**: Spring Security + JWT
* **Flow**:
  + Login → Token → Secure API calls
  + Role-based UI rendering
  + Token refresh and logout handling

### CI/CD

* **Tech**: GitHub Actions
* **Pipeline**:
  + Build and test backend and frontend on push
  + Deploy backend to Render/Railway
  + Deploy frontend to GitHub Pages or Vercel
  + Secrets managed via GitHub repository settings

### Deployment

| **Layer** | **Platform** |
| --- | --- |
| Backend | Render / Railway |
| Frontend | GitHub Pages / Vercel |
| Database | Railway / Supabase (optional) |

## Tech Stack Summary

| **Layer** | **Technology** | **Purpose** |
| --- | --- | --- |
| Frontend | HTML, CSS, JavaScript ( Bootstrap) | UI/UX, client-side logic |
| Backend | Java + Spring Boot | REST APIs, business logic |
| Database | MySQL | Structured data storage |
| ORM | Spring Data JPA / Hibernate | Object-relational mapping |
| API Docs | Swagger / SpringDoc OpenAPI | Auto-generated API documentation |
| Authentication | Spring Security + JWT | Secure login, role-based access |
| CI/CD | GitHub Actions | Automated testing and deployment |
| Deployment | Render (Backend), GitHub Pages / Vercel (Frontend) | Hosting |

## Outcome

* Real-world compliance
* Centralized control for sports/cultural events
* Transparent finance tracking
* Auditable election results
* Browser-first access for students and staff
* Collaborative event management across departments and individuals

## Future Scope

* Biometric login via WebAuthn
* Offline voting via service workers
* Role-based analytics dashboard
* PWA fallback for mobile access
* Cross-campus collaboration for inter-college events
* Dynamic role delegation for collaborative events