

## Optimized 6-Week Plan: Event Registration Website

**Project:** Event Registration Website

**Duration:** 6 Weeks (42 Days)

**Features:** Cultural, Technical, Travel/Adventure, Creative Events, Push Notifications, Chatbot, Admin Dashboard, Analytics

### Week 1: Project Setup & Core Structure

Day	Task
1	Define event categories (cultural, technical, travel/adventure, creative); set up project repo and initial documentation.
2	Create low-fidelity wireframes for homepage and registration flows.
3	Set up backend: initialize Node.js/Express project, MongoDB schema for events/users.
4	Set up frontend: initialize React project, configure routing for event categories.
5	Implement authentication (register/login) backend and frontend UI.
6	Design and implement event creation/listing APIs and basic event cards UI.
7	Internal review: test setup, update documentation, and fix initial bugs.

### Week 2: Event Registration & MVP Homepage

Day	Task
8	Build event registration form UI and connect to backend API.
9	Implement user roles (admin, attendee) and protect routes accordingly.
10	Add event detail pages and registration confirmation logic.
11	Integrate homepage sections for each event category with sample events.
12	Make homepage responsive and accessible.
13	Prepare MVP (minimal viable product) version; conduct internal walkthrough.
14	Deploy MVP to staging server; collect initial team feedback.

### Week 3: Notifications & Chatbot Integration

Day	Task
15	Design push notification UI and backend endpoints (using Firebase or similar).
16	Integrate push notification logic in frontend and backend.
17	Design chatbot UI; decide on chatbot tech (Dialogflow/custom Socket.io).
18	Implement chatbot backend and basic frontend integration.
19	Test notification and chatbot flows; refine UX.
20	Conduct user testing with a small group; gather actionable feedback.
21	Prioritize and address feedback; update documentation.

### Week 4: Admin, Analytics, and Advanced Features

Day	Task
22	Build admin dashboard for event management (CRUD, analytics).
23	Add event filtering, sorting, and search functionality.

24	Integrate analytics (event popularity, registration stats) in admin view.
25	Polish event registration flow; add error handling and loading states.
26	Conduct accessibility and cross-browser testing.
27	Prepare for user demo; update user and API documentation.
28	User/stakeholder demo; collect structured feedback.

#### Week 5: QA, Polish, and Buffer

Day	Task
29	Address demo feedback; fix bugs and optimize performance.
30	Add advanced features (optional): event reminders, calendar integration.
31	Perform security review and code cleanup.
32	Write/expand automated tests (unit, integration, UI).
33	Prepare deployment pipeline (CI/CD, environment configs).
34	Finalize user guides and developer documentation.
35	Internal QA and final review.

#### Week 6: Launch & Post-Launch

Day	Task
36	Deploy production version; verify all features live.
37	Monitor logs, analytics, and user feedback.
38	Hotfix any critical post-launch issues.
39	Conduct a post-mortem meeting; document lessons learned.
40	Plan for future updates (feature backlog, user requests).
41	Team celebration and knowledge sharing session.
42	Buffer day for any spillover or last-minute fixes.