# CS691: Eventify – Project Entry Criteria Presentation Script

## Speaker 1 – Application Introduction - Surmed

(Visual: Title slide or Eventify overview image)

Good evening everyone.

Imagine a world where hosting an event is as simple as a few clicks —

no spreadsheets, no chaos, and no third-party headaches.

That’s exactly what we set out to achieve with **Eventify**

— a smart, all-in-one event and ticketing management platform built to transform how organizers connect with their audiences.

Eventify eliminates the everyday pain points of event management

— juggling multiple tools for registration, ticketing, payments, and promotion.

Instead, it brings everything together in one **seamless, data-driven platform** that saves time, reduces cost, and enhances engagement.

We serve two key groups: **event organizers** and **attendees**.

For organizers, Eventify offers a powerful yet intuitive dashboard

— complete with drag-and-drop event creation, real-time sales tracking, and deep insights into attendee behavior.

For attendees, it’s a smooth and enjoyable experience —

they can discover exciting events, purchase tickets securely, and receive digital QR codes for instant entry.

What truly sets Eventify apart from platforms like Eventbrite or Ticketmaster is our **focus on accessibility and empowerment**.

We prioritize community events, offer **lower commission rates**, and include **built-in marketing tools** that help smaller organizations stand out.

In short, Eventify isn’t just another ticketing platform — it’s the future of event management, built for efficiency, transparency, and growth.

## Speaker 2 – Functional Decomposition & Fishbone Diagram - Surmed

(Visual: Slide from “1. Eventify-FishBone&FuncDecompDiagram.pptx”)

Starting with the Functional Decomposition Diagram, Eventify is structured into five major modules shown in the boxes:

1. Account Management

2. Event Browsing and Discovery

3. Ticketing and Payments

4. Organizer Management

5. Notifications and Communication

Each module represents a logical section of our system and was assigned to a specific team member. For instance, Dhruv led Account Management, Surya with Event Browsing and Discover, Riddhi with Ticketing & Payments, Nand with Organizer Management, and Divy with Notifications & Communications.

Moving to the Fishbone Diagram, you can see how each branch expands these modules into core features. For example:

- Under Account Management, we have registration, login, and authentication.

- Ticketing and Payments includes purchasing, applying promo codes, and generating digital QR tickets.

- Notifications cover reminders, confirmations, and cancellations.

This visualization helped us ensure full feature coverage and proper task assignment before development began.

## Speaker 3 – Context Diagram - Nand

(Visual: “2. Eventify\_Context\_Diagram\_v3.pptx”)

This is our Context Diagram, which illustrates how the Eventify application interacts with both internal and external systems.

At the center, we have the Eventify system itself, which coordinates all data exchanges. Surrounding it are the external entities:

- Event Organizers, who create and manage events;

- Customers, who purchase tickets online; and

- Payment Systems such as PayPal and Stripe, which process transactions securely.

Internally, Eventify connects to several key components:

- The AWS RDS database, which stores all event, user, and transaction information;

- The Notification Service, responsible for sending automated emails and SMS messages; and

- Internal teams such as Finance, Marketing, and Customer Support, who access the system for analytics, promotions, and customer inquiries.

Finally, on the lower right, we have Reporting and Analytics Systems, which use data from AWS RDS to generate sales and attendance reports.

This diagram demonstrates the complete flow of information — from the customer’s ticket purchase to backend processing and internal reporting — showcasing Eventify’s integrated and secure system architecture.

## Speaker 4 – Roadmap - Dhruv

(Visual: “3. RoadMap S1 Eventify\_v2.xlsx”)

Displayed here is our Eventify Roadmap, outlining the major milestones across three development sprints.

In the first column, we have Sprint 1, which focuses on building the application’s foundation — including account creation, event browsing, ticket purchasing, and payment gateway integration. These features establish the complete user flow from registration to digital ticket confirmation.

Moving to Sprint 2, we expand into organizer-focused capabilities such as event management dashboards, capacity checks, and sales analytics.

Finally, Sprint 3 introduces advanced marketing and automation features — referral programs, discount systems, and push notifications — ensuring a complete end-to-end experience for both organizers and attendees.

Each milestone on the roadmap corresponds to deliverables in our RCT and Jenkins pipeline planning, ensuring our schedule remains both agile and outcome-driven. This structured approach allows our team to deliver measurable progress each sprint while maintaining quality and scalability.

## Speaker 5 – RCT / Planning View – Riddhi

(Visual: “4. Eventify-RCT-1012-v2.xlsx”)

Here we have our Requirement Crosscut Table, or RCT, which connects business requirements to their corresponding technical implementations.

If you look across the table, each row represents a user story — for example, ‘User registration,’ ‘Event discovery,’ or ‘Ticket purchase.’ Each column represents a crosscutting concern such as user interface, backend logic, database, security, and DevOps.

For Sprint 1, we prioritized high-value stories that deliver a functional ticketing system — account creation, browsing events, making payments, and generating QR tickets. Each story is color-coded to indicate team responsibility and sprint allocation.

This structure ensures that both functional and non-functional requirements — like security, scalability, and usability — are addressed concurrently. The RCT became our central planning artifact, helping us verify dependencies and maintain transparency across development, testing, and deployment.

## Speaker 6 – Jenkins Progress Report – Thilak/Divy

(Visual: “5. Team1\_Jenkins Progress Report.xlsx”)

Next, we’re showing our Jenkins Progress Report, which outlines the automation pipeline used for Eventify’s development and deployment.

As seen here, Jenkins is fully integrated with our GitHub repository, triggering a pipeline whenever new code is pushed. The report highlights each stage of the pipeline:

- Build, where the system compiles and verifies code;

- Test, where automated checks ensure functionality and stability; and

- Deploy, where Jenkins publishes updates to our AWS EC2 development and QA environments.

You’ll also notice the log entries confirming that the connection between Jenkins and GitHub has been tested successfully. Additionally, our sprint planning section within this document details upcoming tasks for pipeline enhancements and monitoring.

This automation reduces manual effort, speeds up delivery, and ensures version consistency across all environments — aligning with modern CI/CD best practices.

## Speaker 7 – Jenkins Demo - Thilak

(Visual: Pre-recorded Jenkins demo playback)

To conclude our presentation, we’ll show a short two-minute Jenkins demo demonstrating our continuous integration pipeline in action.

In this recording, Jenkins automatically retrieves the latest source code from GitHub, runs a build and test sequence, and deploys the updated application to our AWS development environment. Once deployment completes, Jenkins notifies our team of the successful build status.

This demonstration highlights our readiness for continuous integration and deployment, ensuring faster iteration, higher quality control, and a scalable development process.

That brings our presentation to a close. Thank you for your time and attention.