# HackSphere: Comprehensive System Workflow Report

## 1. Executive Summary

**HackSphere** is an end-to-end lifecycle management platform for hackathons, designed for academic and professional innovation challenges. The system utilizes a role-based access control (RBAC) model to manage Students, Mentors, Judges, SPOCs, and Admins. This report outlines the technical and operational workflows of each role, detailing the interaction between the frontend, backend, and database layers.

## 2. Role-Based Technical Workflows

### 2.1 Student Workflow

The student journey covers everything from initial onboarding to final result viewing.

**Stage 1: Registration and Authentication**

- **Entry Point**: User accesses the `/signup` page and provides personal details along with an Institute Code.
- **Backend Processing**: `authController.signup` hashes the password using Bcrypt and validates the provided Institute Code against the `Institute` registry.
- **Database Interactions**: Creates a new document in the `User` collection.
- **Validation Rules**: Ensures email uniqueness and validates that an active SPOC exists for the selected institute.
- **State Transitions**: Guest user → Registered Student.
- **Notifications**: Confirmation email or dashboard alert upon successful registration.

**Stage 2: Team Creation**

- **Entry Point**: Student clicks "Create Team" on the dashboard sidebar.
- **Backend Processing**: `teamController.createTeam` initializes a new team object, assigning the creator as the `leaderId` .
- **Database Interactions**: Inserts a new `Team` document; updates the User's `teamId` field.
- **Validation Rules**: The user must not already be associated with any team.
- **State Transitions**: Individual Student → Team Leader.
- **Notifications**: Broadcasts a "New Team Created" alert to the Institute's SPOC.

**Stage 3: Problem Selection**

- **Entry Point**: Team Leader navigates to "Problem Statements" and clicks "Select" on a specific problem.
- **Backend Processing**: `teamController.selectProblem` associates the `problemId` with the team.
- **Database Interactions**: Updates the `problemId` in the `Team` collection.
- **Validation Rules**: Team status must be `'approved'` by the SPOC; selection is restricted to the Team Leader.
- **State Transitions**: Approved Team → Team with Assigned Problem.
- **Notifications**: Sends a "Problem Selected" notification to all team members.

**Stage 4: Project Submission**

- **Entry Point**: Team Leader goes to "My Team" page and uploads files/repository URLs.
- **Backend Processing**: `submissionController.createSubmission` saves the file metadata and manages versioning ( `v1` , `v2` , etc.).
- **Database Interactions**: Creates a new `Submission` record linked to the `teamId` .