**Requirements Document: Employee Scorecard & Contribution Tracking Web Application**

**Document Version:** 1.0

**Date:** May 15, 2025

**Prepared By:** Amzad Basha

**1. Introduction**

**1.1. Purpose**

The purpose of this document is to outline the functional and non-functional requirements for a web-based Employee Scorecard and Contribution Tracking application. This application aims to centralize the tracking of employee contributions within the organization, providing a transparent mechanism for recognizing efforts in areas such as code contributions, commits, reviews, session attendance, and session delivery. The goal is to motivate employees, foster a culture of continuous contribution and learning, and provide data-driven insights into team and individual performance.

**1.2. Scope**

This document defines the requirements for the initial release of the Employee Scorecard application, covering user roles, activity submission, approval workflows, point calculation, and basic reporting. Integrations with external HR systems or payroll are out of scope for this initial phase.

**1.3. Audience** This document is intended for project stakeholders, product owners, development teams, quality assurance teams, and future users of the application.

**2. Functional Requirements**

**2.1. User Management & Authentication**

* **REQ-AUTH-001:** The system SHALL allow users to log in with their existing organizational credentials (e.g., SSO, LDAP integration, or internal authentication).
* **REQ-AUTH-002:** The system SHALL support distinct user roles:
  + **Admin:** Full control over activity definitions, points, user roles, and global reporting.
  + **Employee:** Submit activities, view personal scorecard, view status of submissions.
  + **Manager:** Approve/reject submissions from their direct reports, view team scorecards.
  + **Business Head:** Provide final approval for points, view aggregated departmental/organizational scorecards.
* **REQ-AUTH-003:** The system SHALL enforce role-based access control (RBAC) to ensure users can only perform actions relevant to their assigned role.

**2.2. Admin Module**

* **REQ-ADMIN-001:** The Admin SHALL be able to define and manage "Activity Types" (e.g., "Code Commit," "Code Review," "Tech Talk Attended," "Tech Talk Delivered," "PR Reviewed").
* **REQ-ADMIN-002:** The Admin SHALL be able to assign and modify "Points" for each defined Activity Type.
* **REQ-ADMIN-003:** The Admin SHALL be able to define and manage thresholds or tiers for recognition (e.g., "Bronze Contributor" at X points, "Silver" at Y points).
* **REQ-ADMIN-004:** The Admin SHALL be able to add, edit, and deactivate user accounts and assign/change user roles.
* **REQ-ADMIN-005:** The Admin SHALL be able to view a comprehensive dashboard of all activities, pending approvals, and overall organizational scores.

**2.3. Employee Module**

* **REQ-EMP-001:** Employees SHALL be able to submit their activities. For each submission, they SHALL provide:
  + Activity Type (selected from Admin-defined list)
  + Date of Activity
  + Brief Description/Context of the activity
  + Relevant Proof/Link (e.g., GitHub commit link, meeting invite, presentation link, screenshot, document upload)
  + Option to select their Manager for approval.
* **REQ-EMP-002:** Employees SHALL be able to view their personal scorecard, displaying:
  + Total accumulated points.
  + Breakdown of points by activity type.
  + Points earned over specific periods (e.g., weekly, monthly, quarterly, annually).
* **REQ-EMP-003:** Employees SHALL be able to view the status of their submitted activities (e.g., "Pending Manager Approval," "Approved by Manager," "Pending Business Head Approval," "Approved," "Rejected").
* **REQ-EMP-004:** Employees SHALL receive notifications (in-app and/or email) on the status change of their submissions.

**2.4. Manager Module**

* **REQ-MGR-001:** Managers SHALL be able to view a list of all pending activity submissions from their direct reports.
* **REQ-MGR-002:** Managers SHALL be able to review each submission, including the details and provided proof.
* **REQ-MGR-003:** Managers SHALL be able to Approve or Reject a submission.
  + Upon approval, the submission SHALL be forwarded to the Business Head for final approval (if applicable).
  + Upon rejection, the Manager SHALL be required to provide a reason/comment.
* **REQ-MGR-004:** Managers SHALL be able to view the individual scorecards of their direct reports.
* **REQ-MGR-005:** Managers SHALL receive notifications (in-app and/or email) for new submissions from their team members.

**2.5. Business Head Module**

* **REQ-BH-001:** Business Heads SHALL be able to view a list of all activity submissions requiring their final approval.
* **REQ-BH-002:** Business Heads SHALL be able to review each submission, including details, proof, and manager's approval/comments.
* **REQ-BH-003:** Business Heads SHALL be able to provide final Approval or Rejection.
  + Upon final approval, the points for the activity SHALL be awarded to the employee's scorecard.
  + Upon final rejection, the Business Head SHALL be required to provide a reason/comment.
* **REQ-BH-004:** Business Heads SHALL be able to view aggregated scorecards for departments or the entire organization.
* **REQ-BH-005:** Business Heads SHALL receive notifications (in-app and/or email) for submissions awaiting their final approval.

**2.6. Scorecard & Reporting Dashboards**

* **REQ-REP-001:** The system SHALL display individual scorecards with total points and breakdown by activity type.
* **REQ-REP-002:** The system SHALL provide various types of charts to visualize data:
  + **Bar Charts:** Points earned by activity type, points earned per month/quarter.
  + **Line Charts:** Trend of points over time.
  + **Pie Charts:** Distribution of points across different activity categories.
  + **Leaderboards:** Top contributors (e.g., top 10 employees by total points, top X in specific activity categories).
* **REQ-REP-003:** All reporting dashboards SHALL allow filtering by date range, team, department, and activity type.
* **REQ-REP-004:** Data on dashboards SHALL be updated in near real-time upon approval of activities.

**3. Non-Functional Requirements**

**3.1. Performance**

* **NFR-PERF-001:** The application SHALL load dashboards and reports within 3-5 seconds for up to 50 concurrent users.
* **NFR-PERF-002:** Activity submission and approval actions SHALL complete within 2 seconds.

**3.2. Security**

* **NFR-SEC-001:** The application SHALL use HTTPS for all communication.
* **NFR-SEC-002:** User authentication SHALL be secure (e.g., strong password policies, multi-factor authentication if integrated with SSO).
* **NFR-SEC-003:** Role-based access control (RBAC) SHALL be strictly enforced to prevent unauthorized data access or actions.
* **NFR-SEC-004:** All sensitive data (e.g., user credentials) SHALL be encrypted at rest and in transit.

**3.3. Usability (UX/UI)**

* **NFR-USABILITY-001:** The user interface SHALL be intuitive, user-friendly, and easy to navigate.
* **NFR-USABILITY-002:** The application SHALL be responsive, adapting to various screen sizes (desktop, tablet, mobile).
* **NFR-USABILITY-003:** Clear feedback messages SHALL be provided for all user actions (e.g., "Activity submitted successfully," "Approval pending").

**3.4. Scalability**

* **NFR-SCAL-001:** The architecture SHALL be scalable to support a growing number of employees and activities without significant performance degradation.

**3.5. Reliability**

* **NFR-REL-001:** The application SHALL have an uptime of 99.5%.
* **NFR-REL-002:** Data integrity SHALL be maintained across all transactions and updates.

**3.6. Maintainability**

* **NFR-MAINT-001:** The codebase SHALL be well-documented and follow established coding standards for easy maintenance and future enhancements.

**4. Technical Requirements (High-Level)**

* **Platform:** Web-based application.
* **Database:** PostgreSQL (as per previous context).
* **Frontend Technologies (Suggested):** React, Angular, or Vue.js.
* **Backend Technologies (Suggested):** Node.js (Express), Python (Django/Flask), or Java (Spring Boot).
* **Deployment Environment:** Cloud-based (e.g., AWS, Azure, Google Cloud).
* **Version Control:** Git.

**5. Assumptions**

* **ASM-001:** Users will have access to a stable internet connection.
* **ASM-002:** All necessary organizational data (employee IDs, manager mappings, department structures) can be provided or integrated.
* **ASM-003:** Clear definitions and criteria for each activity type (e.g., what constitutes a "code contribution" or a "session delivered") will be provided by the business.
* **ASM-004:** Business Heads are available and willing to provide final approvals within a reasonable timeframe.

**6. Constraints**

* **CON-001:** Initial release functionality is limited to the scope defined in Section 1.2.
* **CON-002:** [Insert specific timeline constraints here, e.g., "Initial MVP release by Q3 2025"].
* **CON-003:** [Insert specific budget constraints here].

**7. Future Enhancements (Out of Scope for Initial Release)**

* **FUT-001:** Integration with HR systems for automated user provisioning and profile updates.
* **FUT-002:** Gamification features (e.g., badges, streaks, public recognition for milestones).
* **FUT-003:** Automated activity tracking (e.g., direct integration with GitHub for commits/PRs, calendar integration for sessions).
* **FUT-004:** Peer-to-peer recognition module.
* **FUT-005:** Mobile application for iOS and Android.
* **FUT-006:** Advanced analytics and predictive insights (e.g., identifying high-potential contributors).