Project Proposal – Team 9

Intelligent Approval Automation System (Auto-Review Agent)

Project Overview

Manual approval workflows in organizations are often slow, inconsistent, and dependent on human availability, creating delays for routine low-risk requests such as room bookings, club permissions, or IT access approvals.

This project proposes an AI-powered auto-review agent that reads approval-related requests (e.g., emails or form submissions), performs risk analysis based on contextual data, and automatically approves low-risk cases while escalating complex cases to administrators only when necessary.

The goal is to streamline routine workflows by reducing administrative bottlenecks through intelligent automation.

Project Scope

- **Must-Haves**

- Request intake system (email/form input)
- Risk analysis engine
- Auto-approval logic for low-risk requests
- Escalation workflow for complex cases
- Admin dashboard
- Authentication system
- Audit logging system

- **Nice-to-Haves**

- AI-generated decision summaries
- Notification system
- Approval history visualization

- Analytics dashboard

Stretch Goals

- Adaptive learning risk classifier
- NLP-based request parsing
- Role-based approval chains

Project Objectives

- Reduce approval response time by **≥60%**
- Achieve **≥90% accuracy** in risk classification
- Maintain system response time **<2 seconds**
- Support concurrent processing of **100+ requests**

Success Metrics

- Auto-approval precision
- Reduction in manual approvals required
- Processing latency
- User satisfaction rating

Specifications

User Interface (UI) Design

- **Platform:** Web Application

  Key Pages:
- Request Submission Page
- Admin Dashboard
- Approval Queue Page
- Activity Log Page

- **User Interaction Elements:**
- Approval/Escalate buttons
- Risk score display
- Status tracking panel
- Form inputs for submission
-

Backend & APIs

- REST API for request submission and approval workflow
- JWT-based authentication
- Role-based access control
- Audit logging
- Event-driven workflow engine (risk → decision → notify)

Data & AI Model

- Dataset Source: Historical approval request logs
- Preprocessing:
    - Keyword extraction
    - Request metadata parsing
    - Requester history analysis
- Model Functionality:
    - Risk classification
    - Contextual decision recommendation
    - Auto-approval for low-risk requests

Tech Stack

- Frontend: React + TailwindCSS
  Backend: FastAPI (Python)
  Database: PostgreSQL

Authentication: Supabase Auth + JWT
Deployment: Vercel
Storage: Supabase

Hardware Requirements

- Standard development laptops
- Cloud hosting environment

Software Requirements

- VS Code
- GitHub
- Figma
- Supabase
- Postman
-

Project Timeline (Gantt Chart or Milestone Table).

Required: Provide a Phase-based schedule with specific tangible deliverables for each phase.

Include specific deadlines for each task. You can add further categories if needed.

Phase	Duration	Tasks			Status/Deliverable
		Front end	Back end	General	
Phase 1	02/15 - 02/28	Define scope, research, and setup			Architecture Diagram
Phase 2	03/01 - 03/15	UI/UX design, data collection, API setup			High-fidelity Figma/Wireframes.
Phase 3	03/16 - 04/05	Development and initial implementation			Functional Alpha (MVP).
Phase 4	04/06 - 04/20	Testing and integration			Beta Version & QA Test Report.

Phase 5	04/21- 05/01	Final testing, deployment, and presentation	Final Demo & GitHub Handover.

Team Leader Rotation

Duration	Team Leader
02/15 - 03/01	Shayan Asif
03/02 - 03/20	Adith Gangalakunta
03/21 - 04/10	Hamere Abate

Project Team

Role	Team Member	Responsibilities
Project Manager	Shayan Asif	Sprint planning
Backend Lead	Adith Gangalakunta	API & workflow logic
Frontend Lead	Hamere Abate	UI development
Database Architect	Luis Jaimes	DB schema
QA Tester	Carlos Ortiz	Testing

Links

- **GitHub Repository:** [Insert Link]
- **Agile Board (Jira/Trello)**: [Insert Link] (Must show active sprints and task assignments.)
- **Design Document:** [Insert Link] (If you are using tools like canva, figma, etc...)