Project Synopsis: Agentic Workflow Automation Builder

Table of Contents

1. Project Overview

- **Project title:** Agentic workflow automation builder
- **Team name & ID:** A4AI & team Id:60
- **Institute / Course:** GLA UNIVERSITY, Mini-Project

- **Date:** 28 Aug 2025

Problem statement: Users often lack programming knowledge to automate workflows, leading to manual, error-prone, and inefficient processes across various business functions. Existing tools may be complex or lack integrated AI assistance.

Goal: To build a no-code, AI-assisted platform that enables users to easily design, execute, and automate workflows using a drag-and-drop interface, thereby reducing human effort and errors.

Value proposition: Effortless workflow design with drag-and-drop blocks, intelligent AI suggestions, real-time monitoring and logs for transparency, and support for multiple industries and use-cases.

2. Scope and Control

2.1 In-scope

- No-code interface where users can design workflows using drag-and-drop blocks.
- Supporting AI-powered suggestions to guide users in building efficient workflows.
- Enabling workflow execution for business automation tasks such as email alerts, approvals, notifications, data entry, and reporting.
- Offering predefined templates for common use-cases like customer support, HR automation, and sales management.
- Allowing import/export of workflows to ensure reusability and easy sharing.
- Providing real-time monitoring and logs for each workflow to ensure transparency and debugging support.
- Scaling to support multiple industries such as education, healthcare, customer service, and business operations.

2.2 Out-of-scope

- Full programming IDE for custom code integration (beyond existing block capabilities).
- Complex, custom third-party integrations not covered by standard API interfaces in v1.
- Advanced data analytics features beyond basic performance tracking.

2.3 Assumptions

- All users have valid college emails.
- Users have basic computer literacy and understanding of their own business processes.
- AI models for suggestions will be adequately trained and performant.
- Third-party services (e.g., email, CRM) have stable APIs for integration.

2.4 Constraints

- Team skills in React.js/Angular, Node.js/Python (FastAPI/Flask) for development.

- AI engine development and integration.

2.5 Dependencies

- Third-party APIs for CRM, ERP, HRMS, and other services for integration.
- NLP and Machine Learning libraries for the AI Engine.
- Database systems (MySQL/PostgreSQL, MongoDB/Firestore).

2.6 Acceptance criteria and signoff

- GIVEN a new user WHEN they access the platform THEN they can successfully create a workflow using drag-and-drop blocks.
- GIVEN a workflow is designed WHEN it is executed THEN real-time logs show each step's progress and status.
- GIVEN a workflow fails WHEN an error occurs THEN the system notifies the user immediately.

3. Stakeholders and Roles

3.1 Stakeholders and RACI

- **Responsible (R):** Individual or team directly performing the task.
- **Accountable (A):** Person ultimately answerable for the correct and thorough completion of the task.
- **Consulted (C):** People whose opinions are sought.
- **Informed (I):** People who are kept up-to-date on progress.

3.2 Team and Roles

- **Role:** primary function (e.g., Product Owner, Frontend Developer, AI Specialist).
- **Responsibilities:** Key tasks and areas of ownership.
- **Contact:** kanishka.gla_cs.aiml23@gla.ac.in

4. Three-Week Plan

Week	Team Member	Assignments	Status
Week 1	KANISHKA	**Frontend:** Initial UI setup and component scaffolding. **Security & Monitoring:** Research best practices for auth and logging. **AI**	Planned

		Engine: Set up AI development environment.	
	ANUSHKA	**Backend:** Set up the project, design initial database schemas. **Data stores:** Configure database connections. **Integrations:** Research third-party APIs.	Planned
Week 2	KANISHKA	**Frontend:** Build the drag-and-drop workflow builder. **AI Engine:** Begin integrating a basic AI suggestion module. **Security & Monitoring:** Implement user authentication.	Planned
	ANUSHKA	**Backend:** Develop core API endpoints for workflows (Create, Read, Update, Delete). **Data stores:** Refine data models. **Integrations:** Set up a mock third-party API connection.	Planned
Week 3	KANISHKA	**Frontend:** Polish UI, test a complete user journey. **Security &**	Planned

		Monitoring: Implement basic logging. **AI Engine:** Finalize AI suggestions.	
	ANUSHKA	**Backend:** Integrate frontend with backend APIs. **Data stores:** Run final migration scripts. **Integrations:** Test data flow through mock connections.	Planned

5. Users and UX

5.1 Top user journeys

- **Workflow Creator:** Home ⟶ Login ⟶ Select Template / Start New ⟶ Drag-and-drop blocks ⟶ Receive AI Suggestions ⟶ Configure Blocks ⟶ Test Workflow ⟶ Deploy. **KPI:** Workflow creation time ≤ 5 minutes for basic flows, AI suggestion adoption rate ≥ 70%.
- **Workflow Monitor:** Dashboard ⟶ View Active Workflows ⟶ Select Workflow ⟶ View Real-Time Logs ⟶ Identify Error ⟶ Receive Alert. **KPI:** Alert delivery time ≤ 10 seconds, error resolution time ≤ 1 hour.

5.2 User stories

EXAMPLE: As an HR Team member, I want to automate daily employee report distribution so that relevant stakeholders receive them consistently every evening at 6 PM.

- **GIVEN:** I am logged in as an HR team member.
- **WHEN:** I configure a scheduled email workflow for daily reports.
- **THEN:** An email containing the employee report is automatically sent to the specified recipients every evening at 6 PM.

5.3 Accessibility & localization

- **Accessibility:** Focus on intuitive UI for visual workflow building, clear error messaging, and adaptable layout for various screen sizes. Keyboard navigation for core interactions.
- **Localization:** Initially English, with potential for future expansion based on market needs.

6. Market and Competitors

- **Competitor table:** Analysis of existing workflow automation platforms (e.g., Zapier, Make,

Microsoft Power Automate, UiPath) detailing their target users, features, pricing, strengths, weaknesses, and key differentiators.
- **Positioning:** Unique selling proposition, focusing on the AI-assisted no-code aspect and cross-industry applicability.
- **Measurable delta:** Quantifiable advantages over competitors (e.g., faster workflow creation, higher automation success rate due to AI).

7. Objectives and Success Metrics

- **O1 User Onboarding:** Achieve a median time-to-first-workflow-creation of under 10 minutes by (specific date). **KPI:** Median minutes.
- **O2 Workflow Execution Reliability:** Maintain a workflow execution success rate of ≥ 98% across all deployed workflows by (specific date). **KPI:** Success rate %.
- **O3 AI Assistance Adoption:** Achieve an AI suggestion acceptance rate of ≥ 60% by (specific date). **KPI:** Acceptance rate %.
- **O4 Scalability & Performance:** Ensure that the platform can handle up to 100 concurrent workflow executions with a p95 latency of ≤ 500 ms for core operations by (specific date). **KPI:** p95 ms.

8. Key Features

Feature	Description	Priority	Dependencies	Acceptance criteria
No-code Workflow Builder	Drag-and-drop interface for designing workflows with Triggers, Conditions, Actions.	Must	Presentation Layer, Application Layer	GIVEN user WHEN drags blocks THEN can connect them visually to form a workflow.
AI-powered Suggestions	Provides intelligent recommendations for next steps and optimizations.	Must	AI Engine, Application Layer	GIVEN a block is placed WHEN user seeks next step THEN relevant AI suggestions appear.
Workflow Execution &	Runs automated	Must	Application Layer,	GIVEN a workflow is

Scheduling	processes based on defined workflows and schedules.		Database Layer	deployed WHEN trigger conditions met THEN it executes as designed.
Predefined Templates	Library of ready-to-use workflows for common business scenarios.	Should	Application Layer, Database Layer	GIVEN user browses templates WHEN selects one THEN a pre-built workflow is loaded.
Import/Export Workflows	Allows users to save and load workflow definitions.	Should	Application Layer	GIVEN a workflow WHEN exported THEN it can be imported and function correctly.
Real-Time Monitoring & Logs	Dashboard and detailed logs for tracking workflow status and performance.	Must	Control & Monitoring Layer, Database Layer	GIVEN a workflow is running WHEN user views dashboard THEN real-time status and logs are visible.
User Auth & Role Management	Secure login, user management, and role-based access control.	Must	Security Layer, Database Layer	GIVEN a user WHEN logs in THEN they only access workflows according to their role.

Error Handling & Alerts	Notifies users immediately of workflow failures and provides guidance.	Must	Control & Monitoring Layer	GIVEN a workflow fails WHEN an error occurs THEN the user receives an alert via chosen channel.

9. Architecture

9.1 High-level

- **Clients:** React.js / Angular SPA (responsive UI, drag-and-drop builder, dashboard, cross-platform access).
- **Services:** Application Layer (Node.js / Python FastAPI/Flask for workflow execution, management, scheduling, third-party API integration).
- **AI Engine:** NLP and ML models for suggestions, optimization, predictive task execution.
- **Data stores:** Relational Database (MySQL/PostgreSQL for user data, workflow definitions, execution logs), NoSQL (MongoDB/Firestore for unstructured AI training logs, recommendations).
- **Integrations:** Third-party APIs (CRM, ERP, HRMS, email services).
- **Security Layer:** User Authentication & Authorization, Data Encryption, Audit Logging, Sandbox Testing.
- **Control & Monitoring Layer:** Real-time dashboard, error logging, performance metrics, alerts.

9.2 Config and secrets

- Environment variables (.env) for local development, excluded from version control (.gitignore).
- Cloud-based secret management for production credentials (API keys for third-party services, database credentials).
- Rotation policies for sensitive credentials.
- Access controls for configuration repositories.

10. Data Design

10.1 Data dictionary

- **User id:** UUID (System, Primary Key)
- **User email:** String (User, Unique, used for authentication)
- **User role:** Enum (admin, manager, staff, System, For Role-Based Access Control)

- **Workflow id:** UUID (System, Primary Key)
- **Workflow name:** String(255) (User, Workflow display name)
- **Workflow definition:** JSON (User, Structured workflow logic)
- **Workflow creatorId:** UUID (System, Foreign Key to User.id, User who created the workflow)
- **WorkflowLog id:** UUID (System, Primary Key)
- **WorkflowLog workflowId:** UUID (System, Foreign Key to Workflow.id, Links to executed workflow)
- **WorkflowLog timestamp:** DateTime (System, Time of log entry)
- **WorkflowLog step:** String(255) (System, Current step in workflow)
- **WorkflowLog status:** Enum (success, failure, pending, System, Status of the step)
- **WorkflowLog message:** Text (System, Detailed log message)

10.2 Schemas and migrations

- Relational database schemas defined for users, workflows, logs.
- Versioned database migrations for schema evolution (e.g., using Flyway or Alembic).
- NoSQL schemas (if used) would be flexible, but with defined access patterns.

10.3 Privacy, retention, backup/DR

- **PII (Personally Identifiable Information):** User email, names (if collected).
- **Data Encryption:** Data at rest and in transit.
- **Retention:** Workflow execution logs and user data retained as per policy (e.g., 1 year for logs, 3 years for inactive user accounts, or as required by GDPR/local regulations).
- **Backups:** Daily incremental backups, weekly full backups for relational databases.
- **DR (Disaster Recovery):** Defined RTO (Recovery Time Objective) and RPO (Recovery Point Objective) with tested recovery procedures.

11. Quality and Compliance

11.1 Quality: NFRs and Testing

- **Dev:** Local development environments for individual team members.
- **Staging:** Pre-production environment for integration testing, UAT, and performance testing.
- **Prod:** Live environment for deployed workflows.
- Feature flags for controlling new functionality releases.
- Sandbox testing environment for workflows before live deployment.

11.2 Security and Compliance

- **Threat model (STRIDE):** Analysis of potential threats to assets.
- **AuthN/AuthZ:** User Authentication (AuthN) and Role-based Access Control (AuthZ).
- **Audit and logging:** Comprehensive logging for all critical actions.
- **Compliance:** Adherence to university policies and general data protection principles.

12. Risks and Mitigations

Risk	Probability	Impact	Score	Mitigation
AI Suggestion Quality	Medium	High	12	Continuous model training, user feedback loops, A/B testing
Integration Complexity	Medium	Medium	9	Standardized API connectors, modular design, thorough testing
Performance Bottlenecks	Medium	High	12	Load testing, optimized database queries, scalable architecture
Security Vulnerabilities	Medium	High	12	Regular security audits, secure coding practices, vulnerability scanning
User Adoption	Medium	Medium	9	Intuitive UX, comprehensive documentation, user training

13. Research and Evaluation

- **Market review:** Analysis of existing no-code automation platforms and AI-assisted tools

to understand best practices, user expectations, and competitive landscape.
- **Evaluation plan:** Tracking key performance indicators (KPIs) outlined in Section 7. User surveys and feedback collection after initial deployment.
- **Limitations:** Initial version may have limited third-party integrations, AI suggestions might not cover all edge cases, and scalability will be progressively enhanced.

14. Appendices

14.1. References

- **React.js Documentation:** https://react.dev/
- **Angular Documentation:** https://angular.io/docs
- **Node.js API Reference:** https://nodejs.org/docs/latest/api/
- **Python (FastAPI) Docs:** https://fastapi.tiangolo.com/
- **Tailwind CSS Docs:** https://tailwindcss.com/docs
- **Firebase/Firestore Docs:** https://firebase.google.com/docs