

Every successful startup begins with an idea — but only those with the right guidance, structure, and support survive. Our project, *Tasavur*, is built to provide exactly that

1. PROBLEM STATEMENT:

1. Manual startup onboarding
2. No centralized mentorship tracking
3. Limited visibility for investors

2. PROJECT OBJECTIVES:

- Build a centralized digital incubator platform
- Enable role-based access for all stakeholders
- Automate mentorship and funding workflows

3. SYSTEM USERS (ROLES)

- Admin
- Founder
- Mentor
- Investor

4. TECHNOLOGY STACK

Frontend

- React.js
- Tailwind CSS

Backend

- Node.js
- Express.js

Database

- MongoDB (Mongoose)

Other Tools

- REST APIs (A REST API allows the frontend to **request data or send data** to the backend using standard rules. Common REST API Methods **GET** → Fetch data, **POST** → Send new data, **PUT** → Update data, **DELETE** → Remove data)
- JWT Authentication (**JWT authentication is a secure way to verify a user without storing session data on the server. How it works**)
- User logs in with email and password
- Server verifies details and generates a **JWT token**

- Token is sent to the client
- Client sends this token with every request
- Server checks the token and allows or denies access

5. Animation:

- **No external animation libraries** (no Framer Motion, GSAP, Lottie)
- Pure **SVG + CSS keyframes + React state**
- **SVG filters** for glow/blur effects

What it is

SVG (Scalable Vector Graphics) is an **image made of code**, not pixels.
It stores **shapes + math**, not dots.

How it works

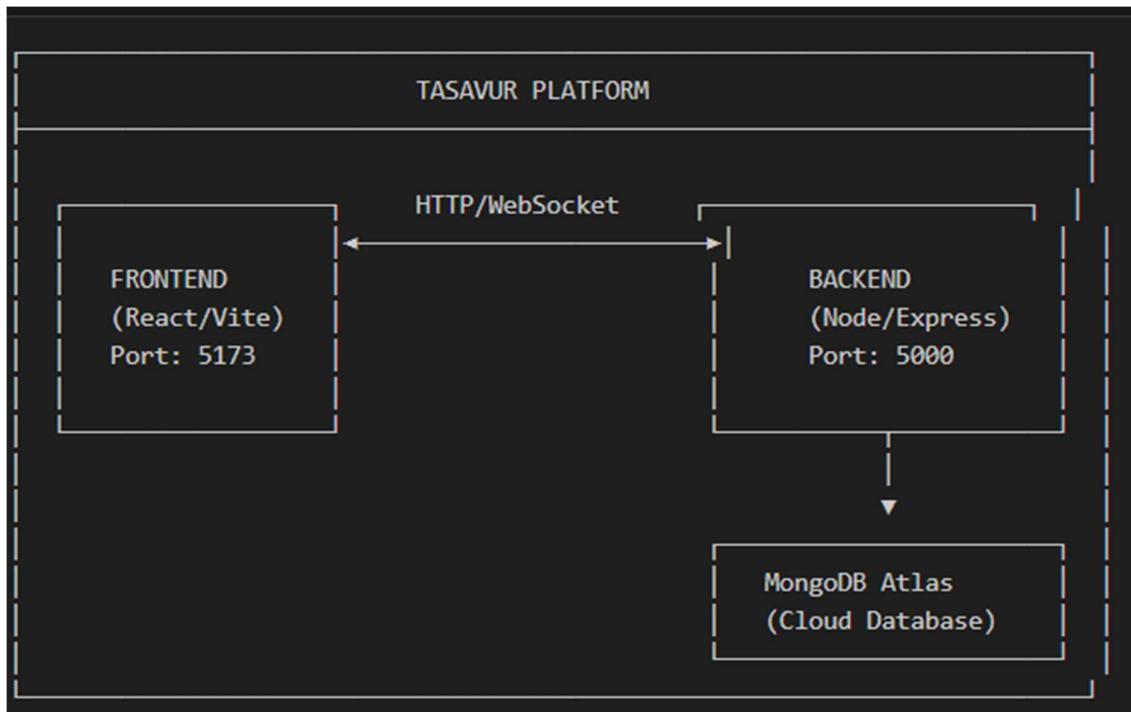
1. The browser reads **SVG code (XML)**
2. That code describes **lines, curves, circles, paths**
3. The browser **draws the image live** using math
4. When you zoom → the math recalculates → **still sharp**

No stretching. No blur. Just precision.

What's inside an SVG

- **Coordinates (x, y)**
- **Shapes** (circle, rect, path)
- **Styles** (color, stroke, fill)
- Optional **animation & interaction**

Complete Technical Overview:



Technology Stack

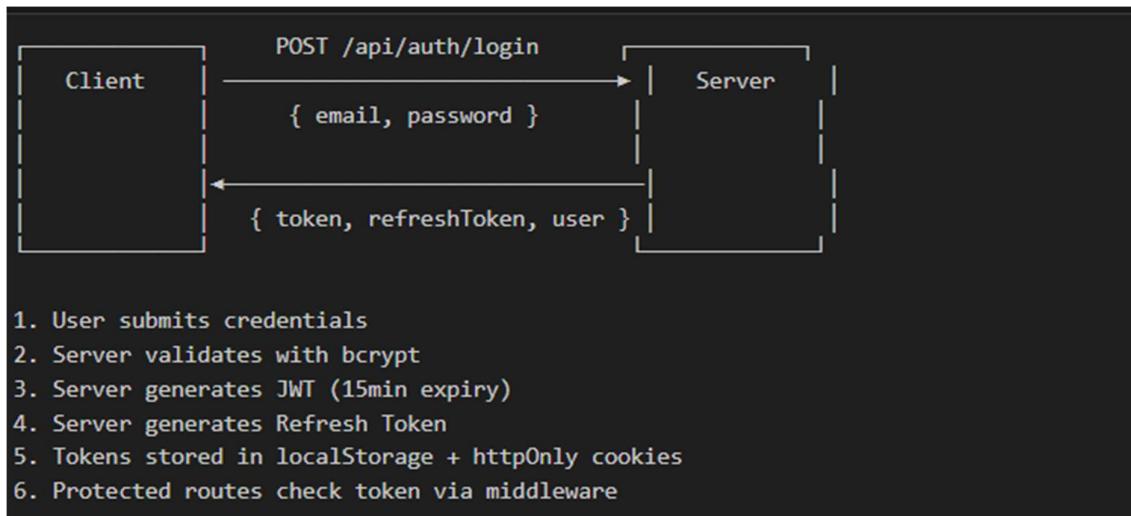
Backend (Node.js/Express):

Technology	Purpose
Express.js	Web framework for building REST APIs
MongoDB + Mongoose	NoSQL database and Object Data Modeling (ODM)
JWT	Authentication using access and refresh tokens
bcryptjs	Secure password hashing
Socket.IO	Real-time bidirectional communication
Multer + AWS S3	File uploads (pitch decks, profile images)
Helmet	Adds security-related HTTP headers
express-rate-limit	Protects against brute-force attacks
express-validator	Input validation and sanitization
cookie-parser	Handles cookies for authentication tokens

Frontend (React / Vite)

Technology	Purpose
React 18	Component-based UI library
Vite	Fast development server and build tool
React Router v6	Client-side routing
Zustand	Lightweight state management
Axios	HTTP client for API communication
Tailwind CSS	Utility-first CSS framework
Headless UI	Accessible UI components
React Hot Toast	Toast notifications
Socket.IO Client	Real-time updates

Authentication Flow:



JWT Token Structure:

- **Access Token:** 15 minutes expiry
- **Refresh Token:** Longer-lived, stored in database

Role-Based Access Control (RBAC)

Role	Access Level	Key Features
Admin	Full Access	Manage users, approve/reject startups, manage resources, dashboards
Founder	Startup Owner	Create startup, request mentorship, apply for funding
Mentor	Advisory Role	View and manage mentorship requests, track sessions
Investor	Funding Access	Browse startups, review funding applications

AI-Powered Mentor Matching Algorithm

The mentor matching system uses a **weighted scoring mechanism** to rank mentors.

Scoring Criteria

- **Skill Match (0-100)**
 - Exact match → full score
 - Partial match → half score
- **Domain Match (0-100)**
 - Based on industry overlap
- **Availability Score (0-100)**
 - Available → 100
 - Busy → 50
 - Unavailable → 0
- **Rating Score (0-100)**
 - Mentor rating: up to 80 points
 - Experience bonus: 1 point per session (max 20)
- **Capacity Score (0-100)**
 - Higher score for mentors with available slots

Optional Enhancement:

Semantic matching using AI embeddings for deeper skill and description analysis.

Real-Time Features (Socket.IO)

- Live dashboard updates
- Instant mentorship request notifications
- Real-time status changes

Database Models:

Model	Purpose
User	Authentication, roles, profiles
Startup	Startup details, stage, funding, KPIs
Mentor	Skills, availability, ratings
MentorshipRequest	Session requests and tracking
FundingApplication	Investor funding requests
Resource	Educational content and documents
Notification	In-app notifications

Security Features:

Feature	Implementation
Password Hashing	bcrypt with salt rounds
JWT Security	Short-lived access tokens + refresh rotation
Rate Limiting	100 requests per 15 minutes per IP
Helmet	XSS, clickjacking, security headers
CORS	Whitelist-based origin control
Input Validation	express-validator middleware
Environment Validation	Required env variables checked on startup

API Endpoints:

Route	Purpose
/api/auth/*	Authentication (login, register, refresh, logout)
/api/startup/*	Startup CRUD operations
/api/mentorship/*	Mentorship requests and matching
/api/funding/*	Funding applications
/api/resource/*	Resource management
/api/dashboard/*	Admin dashboard analytics

