

Report On

**“StartUp Ayush Portal”**

**Team details**

| **Sl. no.** | **Roll no.** | **Name** |
| --- | --- | --- |
| **2.** | **20211CSG0021** | **Pawan P** |
| **3.** | **20211CSG0022** | **Rishith R Rai** |
| **5.** | **20211CSG0031** | **Ganashree P** |

**School of Computer Science ,**

**Presidency University, Bengaluru**

**Under the Supervision of**

**Dr. Saravana Kumar (Associate Professor)**

**School of Computer Science and Engineering, Presidency University.**

**Abstract**

The Startup AYUSH Portal is a comprehensive digital platform designed to foster innovation, collaboration, and entrepreneurship in the AYUSH (Ayurveda, Yoga & Naturopathy, Unani, Siddha, and Homeopathy) sector. This one-stop platform connects startups, investors, incubators, accelerators, government agencies, and the public, enabling seamless interaction in a dynamic ecosystem.

The portal provides a collaborative environment for networking, mentorship, funding opportunities, and resource sharing. Entrepreneurs in the AYUSH domain can showcase their innovations, access government schemes, and connect with key stakeholders globally. Additionally, the platform offers virtual events, educational resources, and incubation support, ensuring the sustainable growth of AYUSH-based startups.

By integrating technology with traditional healthcare systems, the Startup AYUSH Portal aims to accelerate the global expansion of AYUSH practices, promote wellness-based startups, and support research and development in the field. Through a user-friendly digital interface, it bridges the gap between innovation and investment, making AYUSH-based entrepreneurship more accessible and impactful.

**Introduction**

The Startup Ayush Portal is a cutting-edge, digital platform designed to revolutionise the Ayush(Ayurveda, Yoga & Naturopathy, Unani, Siddha, and Homeopathy) ecosystem by fostering innovation, collaboration, and growth. Built on the MERN stack (MongoDB, Express.js, React.js, and Node.js), this portal leverages modern web technologies to deliver a seamless, scalable, and user-friendly experience for all stakeholders, including startups, investors, incubators, accelerators, government agencies, and public users. The Ayush sector, with its deep roots in traditional medicine and wellness practices, has immense potential to contribute to global healthcare and economic development.

However, startups in these sectors often face unique challenges, such as limited access to funding, mentorship, regulatory guidance, and global markets. Existing platforms are either generic or lack the specialised features needed to address these challenges, creating a gap in the ecosystem. The Startup Ayush Portal addresses these gaps by providing a dedicated, one-stop platform tailored to the needs of the Ayush community. By harnessing the power of the MERN stack, the portal offers a robust, dynamic, and interactive environment where stakeholders can connect, collaborate, and thrive. Key features include stakeholder dashboards, a centralized resource repository, virtual networking tools, a global showcase platform, and data-driven analytics, all designed to empower Ayush startups in their entrepreneurial journey.

The use of the MERN stack ensures that the portal is not only highly performant and scalable but also equipped with modern UI/UX capabilities to enhance user engagement. MongoDB provides a flexible and scalable database solution, Express.js enables efficient backend development, React.js ensures a responsive and interactive frontend, and Node.js facilitates real-time communication and seamless integration of features.

By bringing together the entire Ayush startup community on a single platform, the Startup Ayush Portal aims to create a vibrant, interconnected ecosystem that drives innovation, supports collaboration, and unlocks the full potential of the Ayush sector. Whether it's connecting startups with investors, providing access to mentorship and resources, or showcasing innovations to a global audience, this portal poised become the cornerstone of the Ayush startup ecosystem, both in India and worldwide.

**Pseudo Code**

// backend/server.js

const express = require('express');

const cors = require('cors');

const mongoose = require('mongoose');

const authRoutes = require('./routes/auth');

const startupRoutes = require('./routes/startups');

const app = express();

app.use(cors());

app.use(express.json());

mongoose.connect(process.env.MONGO\_URI, { useNewUrlParser: true, useUnifiedTopology: true });

app.use('/api/auth', authRoutes);

app.use('/api/startups', startupRoutes);

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

// backend/routes/auth.js

const express = require('express');

const router = express.Router();

const jwt = require('jsonwebtoken');

const User = require('../models/User');

router.post('/login', async (req, res) => {

const { email, password } = req.body;

// Normally you'd hash and compare passwords

const user = await User.findOne({ email });

if (!user || user.password !== password) return res.status(401).json({ error: 'Invalid credentials' });

const token = jwt.sign({ id: user.\_id, role: user.role }, process.env.JWT\_SECRET);

res.json({ token });

});

module.exports = router;

// frontend/src/pages/Login.jsx

import { useState } from 'react';

import axios from 'axios';

function Login() {

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const handleLogin = async () => {

try {

const res = await axios.post('http://localhost:5000/api/auth/login', { email, password });

localStorage.setItem('token', res.data.token);

// Redirect or update auth context

} catch (err) {

alert('Login failed');

}

};

return (

<div className="max-w-sm mx-auto mt-10 p-6 bg-white shadow-lg rounded-2xl">

<h2 className="text-xl font-bold mb-4">Login</h2>

<input className="w-full p-2 mb-2 border rounded" placeholder="Email" onChange={(e) => setEmail(e.target.value)} />

<input className="w-full p-2 mb-4 border rounded" type="password" placeholder="Password" onChange={(e) => setPassword(e.target.value)} />

<button onClick={handleLogin} className="w-full bg-blue-600 text-white py-2 rounded hover:bg-blue-700">Sign In</button>

</div>

);

}

export default Login;

// frontend/src/App.jsx

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

import Login from './pages/Login';

import Dashboard from './pages/Dashboard';

function App() {

return (

<Router>

<Routes>

<Route path="/" element={<Login />} />

<Route path="/dashboard" element={<Dashboard />} />

</Routes>

</Router>

);

}

export default App;

**Objectives**

This project's main goal is to create a comprehensive startup funding and mentoring platform that connects funding organisations, seasoned mentors, and aspiring entrepreneurs. The system is designed to offer an integrated solution where administrators can effectively manage users and assess applications, and startups can register their ventures, apply for funding, and connect with mentors for guidance.

In order to accomplish this, the site is constructed on a MERN (MongoDB, Express.js, React.js, Node.js) stack that provides frontend and backend features. The backend is implemented on Express.js and Node.js that exposes RESTful APIs to manage user authentication, authorization, and CRUD operations for startup and application data at startup and application levels. MongoDB is the database used that offers a dynamic NoSQL schema to support various data associated with users, startups, mentors, and applications.

The frontend is designed using React.js for building dynamic user interfaces and Tailwind CSS for utility-first styling, ensuring a responsive and modern experience across devices. Additionally, Lucide icons are utilized to enhance the visual clarity and navigation of the platform. Role-based dashboards are implemented to cater to different types of users — startups can track their applications, mentors can manage mentoring requests, and admins can review and approve applications through an intuitive interface.

**Literature Survey**

### 1. Digital Startup Funding Platforms:

### Startup funding has evolved significantly with the introduction of digital solutions that aim to streamline the funding process. Crowdfunding platforms like Kickstarter and Indiegogo have successfully demonstrated the viability of raising funds for various projects. According to Mollick (2014), these platforms depend heavily on transparent communication, community engagement, and effective interaction to meet funding goals. However, crowdfunding tends to be more product-focused and may not meet the needs of all types of startups, particularly those looking for long-term business development rather than just product sales.

* **Key Insight**: The application of digital tools for automating funding applications is a significant step forward in streamlining the process. Traditional venture capital (VC) and angel investing platforms still suffer from inefficiencies in terms of structured application processes and tracking.
* **Literature Insight**: Our platform’s proposed solution for structured role-based interfaces and automated funding application tracking addresses this gap, ensuring that both founders and investors have a seamless, transparent process.

### 2. Mentorship and Entrepreneurial Growth:

### Mentorship is widely considered a cornerstone of entrepreneurial success. In the context of startups, having a mentor can significantly impact decision-making and business strategies. According to St-Jean and Audet (2012), effective mentorship helps entrepreneurs navigate early-stage challenges, sharpen their decision-making, and improve business models.

* **Key Insight**: Digital mentorship platforms, such as MicroMentor and GrowthMentor, have attempted to address this need by connecting founders with seasoned professionals. However, a common challenge in these platforms is the lack of effective mentor matching algorithms that pair mentors with startups based on specific expertise areas.
* **Literature Insight**: To resolve these issues, our platform employs a MongoDB-based filtering system that allows startups to search for mentors based on expertise tags, ensuring more relevant and effective mentor-mentee relationships. This system improves the mentor matching process and promotes more productive interactions.

### 3. Full-Stack Web Solutions in Entrepreneurial Platforms:

Recent advancements in web technologies have enabled the creation of scalable and responsive platforms for startups. The MERN stack (MongoDB, Express.js, React.js, Node.js) has become a popular choice for developing modern, high-performance applications. According to Madhusudan and Kumar (2021), NoSQL databases like MongoDB are particularly well-suited for storing dynamic user-generated content, which is common in entrepreneurial ecosystems where data is constantly changing.

* **Key Insight**: The use of React.js in the frontend allows for dynamic, real-time updates, while Node.js and Express.js provide a lightweight, asynchronous backend for handling requests efficiently.
* **Literature Insight**: Our platform leverages the MERN stack to ensure scalability and maintain performance under high traffic loads. Additionally, JWT-based authentication ensures secure access control, safeguarding sensitive information like funding applications and mentor data.

### 4. Community-Driven Ecosystem Portals:

Community-driven portals have demonstrated strong engagement by allowing users to create personalized profiles and interact with others based on shared interests or goals. Platforms like GitHub, Stack Overflow, and Startup India Hub have fostered vibrant online communities that support peer-to-peer interaction and knowledge-sharing. Research into these communities shows that personalized content and tailored user dashboards enhance engagement.

* **Key Insight**: Personalizing user experiences by tailoring dashboards to each user’s role is critical for maintaining user engagement. Founders, mentors, and administrators all have different needs and interactions with the platform, and a role-based dashboard significantly improves the usability and relevance of the platform.
* **Literature Insight**: By providing dynamic, role-specific dashboards, the platform enhances user engagement and ensures that each user has quick access to the features and information that are most pertinent to them.

### 5. Secure and Scalable Platforms

In today’s digital world, ensuring the security and scalability of platforms is paramount. As the number of users increases, platforms must be able to handle increased traffic and user-generated content without compromising performance. Moreover, secure authentication mechanisms such as JWT (JSON Web Tokens) are necessary to protect user data and prevent unauthorized access.

* **Key Insight**: Scalable backend architectures, especially those built on technologies like Node.js, provide asynchronous and non-blocking processing, which is ideal for handling multiple concurrent requests.
* **Literature Insight**: The platform’s architecture incorporates both scalability and security, using modern web technologies and secure JWT-based authentication to ensure that sensitive user data, such as funding applications and mentor profiles, remains protected.

### 6. Real-Time Updates and Dynamic User Interactions:

### Modern web applications, especially those in the startup ecosystem, benefit from real-time updates to foster dynamic interactions. Features such as notifications, application updates, and messaging between users need to be integrated into the platform to enhance user engagement and operational efficiency.

* **Key Insight**: The use of React.js ensures that the frontend can update dynamically in response to changes in application data, providing a seamless user experience.
* **Literature Insight**: The real-time features, enabled by React and supported by the backend built with Node.js, enable users to interact in real-time, ensuring that communication between founders and mentors is efficient and up-to-date.

### 7. User Experience and Interface Design:

### Effective user interface (UI) and user experience (UX) design are crucial for any platform aimed at user engagement. Research shows that platforms that are intuitive, easy to navigate, and provide personalized content are more successful in retaining users. For example, Slack and LinkedIn have mastered the art of delivering relevant content and ensuring seamless interactions for their users.

* **Key Insight**: By integrating role-specific dashboards and a clean, responsive UI with Tailwind CSS, the platform ensures that users have a pleasant and intuitive experience, which is essential for driving adoption and sustained engagement.
* **Literature Insight**: Tailoring the UI to the specific needs of users (founders, mentors, administrators) maximizes usability and encourages frequent use of the platform.

**Research Gaps of Existing Methods**

1. Lack of Structured Role-Based Interfaces:
   * Most existing platforms like Kickstarter or MicroMentor offer generalized interfaces, without distinguishing between user roles such as startup founders, mentors, and administrators.
   * Gap: Absence of role-specific dashboards leads to cluttered UX and inefficient user journeys.
2. Inadequate Mentor Matching Algorithms:
   * Platforms often use basic or manual methods to connect mentors and startups.
   * Gap: Lack of intelligent, filter-based matching by domain expertise, experience, or startup stage hinders effective mentorship.
3. Limited Application Tracking Capabilities:
   * Many funding platforms don’t provide real-time tracking or status updates for funding applications.
   * Gap: Startups are left in the dark post-submission, reducing transparency and trust in the process.
4. Scalability and Data Management Issues:
   * Legacy systems or SQL-based platforms struggle to handle dynamic user-generated data efficiently.
   * Gap: Inability to scale with growing user base or real-time interactions due to rigid data structures.
5. Security Shortcomings in User Access:
   * Some platforms lack robust authentication and authorization systems.
   * Gap: Absence of secure JWT-based authentication leaves data vulnerable to unauthorized access.
6. Lack of Integration with Third-Party Tools:
   * Existing platforms often operate in silos, lacking integration with tools like LinkedIn or Slack.
   * Gap: Missed opportunity for enriching user profiles, networking, or communication efficiency.
7. Non-Personalized Content and Notifications:
   * Many systems do not tailor notifications or dashboard content to user roles or preferences.
   * Gap: Leads to disengagement due to irrelevant content and lack of personalization.
8. Weak Real-Time Communication Features:
   * Mentorship platforms typically rely on asynchronous messaging or email.
   * Gap: No support for real-time chat or meeting scheduling reduces immediacy and user satisfaction.
9. Minimal Administrative Control & Insights:
   * Current systems rarely empower admins with analytics, user monitoring, or content moderation.
   * Gap: Admins lack tools to manage platform health, detect misuse, or gather insights.
10. Outdated UI/UX and Limited Responsiveness:
    * Some platforms still follow traditional web design patterns with slow load times and unresponsive layouts.
    * Gap: Poor frontend frameworks result in subpar user experience, especially on mobile devices.

(i). **Proposed Solutions:**

* Role-Based Dashboards – Custom interfaces for startups, mentors, and admins improve usability and access to relevant tools.
* Mentor Matching System – Startups can search and connect with mentors using expertise tags and filters powered by MongoDB.
* Application Tracking – Real-time funding application updates and notifications enhance transparency.
* Secure Authentication – JWT-based login ensures role-specific access and user data security.
* MERN Stack Architecture – Utilizes MongoDB, Express.js, React, and Node.js for a modular, scalable, and responsive platform.
* Third-Party Integration – Supports LinkedIn and Slack integration for better communication and user data enrichment.
* Admin Tools – Admins can monitor activity, manage users, and analyze platform performance.
* Responsive UI – React.js enables a dynamic, mobile-friendly interface for seamless user experience.

**Methodology**

The development of the full-stack startup funding and mentorship platform follows a structured approach using agile methodology. The core process includes the following phases:

1. Requirement Analysis:
   * Identified the key users: founders, mentors, and admins.
   * Defined core features: registration, login, role-based access, mentor search, funding application management, and dashboards.
2. System Design:
   * Designed the architecture using the MERN stack (MongoDB, Express.js, React.js, Node.js).
   * Created schema for users, applications, and mentorships.
   * Established frontend routes (React) and backend APIs (Express.js) for each function.
3. Frontend Implementation:
   * Used React.js and Tailwind CSS to build responsive user interfaces.
   * Developed components for login, registration, dashboard views, mentor list, and application forms.
   * Implemented JWT-based session management and role-based rendering.
4. Backend Development:
   * Created RESTful APIs with Express.js for authentication, mentor filtering, and application handling.
   * Used MongoDB to store and manage dynamic user data and application records.
   * Applied authentication and role validation middleware to secure routes.
5. Integration and Testing:
   * Integrated frontend with backend APIs.
   * Tested flows like user registration, login, mentor search, and funding applications.
   * Validated role-specific dashboards and secure access.
6. Deployment and Future Enhancements:
   * Platform is designed to be scalable and deployable on cloud services like Vercel and Render.
   * Future scope includes integrating third-party tools (e.g., Slack, LinkedIn) and adding chat, notification, and advanced filtering systems.

**Conclusion**

The proposed full-stack platform addresses key challenges in startup funding and mentorship by leveraging the MERN stack (MongoDB, Express.js, React.js, Node.js) for a scalable, efficient, and secure solution. The platform enables startups to easily connect with mentors based on expertise tags, improving mentorship quality and interaction. The automated funding application system simplifies the funding process, offering transparency and tracking features that enhance communication between startups and investors.

By incorporating role-based authentication and personalized dashboards, the platform ensures a user-friendly experience for founders, mentors, and administrators. It fills critical gaps in existing funding and mentorship systems, providing an integrated solution that enhances startup growth. The platform’s modular design allows for future improvements and third-party integrations, ensuring its adaptability and long-term success in supporting entrepreneurs.

**Project Timeline**

The project will be executed in four phases to ensure systematic progress toward final implementation and report submission.

Phase 1: Research & Planning (Feb 1 - Feb 15)

This phase involves finalizing the title, conducting a literature survey, defining the problem statement, and planning the methodology. The abstract and initial project timeline will be prepared.

Phase 2: Algorithm Design & Partial Implementation (Feb 16 - Mar 10)

The core algorithms and system architecture will be designed, followed by the development of 50% of the implementation. Initial testing and debugging will ensure smooth progress.

Phase 3: Full Implementation & Report Drafting (Mar 11 - Mar 31)

This phase includes completing the remaining implementation, integration testing, and performance optimization. The first draft of the project report will be prepared and reviewed.

Phase 4: Report Submission & Viva Preparation (Apr 1 - Apr 15)

The final project report will be submitted, followed by mock viva preparation. The final presentation will be created, with potential for publications if applicable.

Key Milestones:

Feb 15: Objectives & research completed

Mar 10: 50% implementation done

Mar 31: Full implementation & report draft ready

Apr 15: Final report submission & viva preparation

**References**

1. A. Gawer, "Bridging differing perspectives on technological platforms," Strategic Management Journal, vol. 35, no. 10, pp. 1319–1337, 2014.

2. E. Stam, "Entrepreneurial ecosystems and regional policy," Research Policy, vol.44, no. 4, pp. 805–816, 2015.

3. Ministry of AYUSH, Government of India, "AYUSH Sector: Opportunities andChallenges," 2021.

4. R. Kumar and A. Sharma, "Entrepreneurship in the AYUSH sector: A study of challenges and opportunities," Journal of Ayurveda and Integrative Medicine, vol.9, no. 4, pp. 308–314, 2018.

5. P. Thakkar, "Building scalable web applications with the MERN stack," Journal of Web Development, vol. 5, no. 2, pp. 45–60, 2020.

6. NITI Aayog, "Strengthening the startup ecosystem in India," 2018.

7. B. Patwardhan, D. Warude, P. Pushpangadan, and N. Bhatt, "Ayurveda and traditional Chinese medicine: A comparative overview," Evidence-Based Complementary and Alternative Medicine, vol. 2, no. 4, pp. 465–473, 2005.

8. F. Provost and T. Fawcett, Data Science for Business. O'Reilly Media, 2013.

9. S. M. Hackett and D. M. Dilts, "A systematic review of business incubation research," The Journal of Technology Transfer, vol. 29, no. 1, pp. 55–82, 2004.

10.K. Chodorow, MongoDB: The Definitive Guide. O'Reilly Media, 2013.