

Introduction	1
System Architecture	2
Overview	2
Communication Flow	2
Technologies Used	2
Installation & Configuration.....	3
Prerequisites.....	3
Clone the Repository	3
Backend Setup (Express.js)	3
Frontend Setup (Vue.js)	3
API Routes: (endpoints to create, read, update and delete the data)	4
Contributors	9

Introduction

JEB Incubator is a web application designed to manage startups and facilitate connections between startups and investors. The platform allows startups to showcase their projects, while investors can discover and engage with innovative ventures.

The system is overseen and managed by administrators, who ensure proper operation, validate content, and maintain the integrity of interactions on the platform.

The main goals of JEb Incubator are:

- **Startup management:** Enable startups to create, update, and present their projects.
- **Investor relations:** Provide investors with tools to browse, evaluate, and connect with startups.
- **Project visibility:** Highlight promising projects to increase their exposure and chances of funding.
- **Administrative control:** Allow administrators to monitor activities, validate accounts, and ensure compliance.

System Architecture

The **JEB Incubator** application follows a **three-tier architecture** consisting of a frontend, a backend, and a database layer.

Overview

- **Frontend (Client-side):** Developed with **Vue.js**, it provides the user interface for startups, investors, and administrators. The frontend communicates with the backend via RESTful API calls.
- **Backend (Server-side):** Implemented using **Node.js with Express.js**, it handles business logic, API endpoints, authentication, and communication with the database.
- **Database:** The application uses **MongoDB** as a NoSQL database to store data related to startups, investors, projects, and admin management.

Communication Flow

1. The **user** interacts with the **Vue.js frontend** through a web browser.
2. The frontend makes **HTTP requests** (REST API) to the **Express.js backend**.
3. The backend processes the request, applies business rules, and interacts with **MongoDB** for data persistence.
4. The backend sends back a **JSON response** to the frontend.
5. The frontend updates the **UI dynamically** based on the response.

Example:

- A **startup** submits a new project → request goes from Vue.js → Express backend validates & stores it → MongoDB saves the data → confirmation is sent back to the UI.

Technologies Used

- **Frontend framework:** Vue.js 3 (with Vue Router & Vuex for state management)
- **Backend framework:** Express.js (Node.js)
- **Database:** MongoDB
- **Package manager:** npm
- **Other tools:**
 - GitHub Actions (CI/CD pipeline)

Installation & Configuration

Prerequisites

Before running the application, make sure you have the following installed:

- **Node.js:** v18+
- **npm:** v9+ (comes with Node.js)
- **MongoDB:** v6+ running locally or on a remote server

Clone the Repository

- `git clone https://github.com/your-username/jeb-incubator.git`
- `cd jeb-incubator`

Backend Setup (Express.js)

Start the backend server at the root of the repository:

- `npm run start`

Backend dependencies (from package.json):

- `express` → Backend framework
- `mongodb` → MongoDB native driver
- `dotenv` → Load environment variables
- `bcrypt` → Password hashing
- `jsonwebtoken` & `jwt-decode` → Authentication (JWT)
- `cors` → Cross-origin resource sharing

Frontend Setup (Vue.js)

Start the frontend server at the root of the repository:

- `npm run dev`

Frontend dependencies (from package.json):

- `vue` → Vue.js 3 framework
- `vue-router` → Client-side routing

- axios → HTTP requests
- @fullcalendar/vue3 & fullcalendar → Calendar integration
- jsonwebtoken & jwt-decode → Token handling for authentication

API Routes: (endpoints to create, read, update and delete the data)

We developed a CRUD API where it serves to communicate between the backend and the frontend. For store the data we used NoSQL (no structured query language) with MongoDB. We decided to use NoSQL because it lets us modify fields of database without deleting them.

/startups:

endpoint	method	Request header	Request body	Request query	access	response
‘/’	GET	X	X	X	everyone	startups
‘/filter’	GET	X	X	Sector, maturity, loc	everyone	Filtered startups
/:id	GET	X	X	Starupd id	everyone	startup
/:id/projects	GET	X	X	Startup id	everyone	Startup projects
/:id/channels	GET	Authorizatio n	X	Startup id	Founder	Startups channels

/news:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	X	X	X	everyone	news
/:id	GET	X	X	News id	everyone	The news

/events:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	X	X	X	everyone	events
/:id	GET	X	X	Event id	everyone	event

/users:

DOCUMENTATION – JEB Incubator

endpoint	method	Request header	Request body	Request query	access	response
/	GET	Authorization	X	X	admin	users
/:id	GET	Authorization	X	User id	admin	user
/login	POST	X	Email, password	X	everyone	token

/projects:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	X	X	X	everyone	projects
/	POST	Authorization	Name, description, sector, maturity	X	admin	Request body
/:id	GET	(Authorization)	X	Project id	everyone	project
/:id	PUT	Authorization	Name, description, sector, maturity	X	admin	Good message
/:id	DELETE	Auhorization	X	Project id	admin	Good message
‘/filter’,	GET	X	X	Sector, maturity, loc	everyone	Filtered projects

/filters:

endpoint	method	Request header	Request body	Request query	access	response
/sector	GET	X	X	X	everyone	All sector available
/sector	PUT	Authorization	Sector name	X	admin	Add a new sector

/channels:

endpoint	method	Request header	Request body	Request query	access	response
----------	--------	----------------	--------------	---------------	--------	----------

DOCUMENTATION – JEB Incubator

/	POST	Authoriz ation	Starup id & name, investor id & name	X	investor	Good message
/:id/cha ts	GET	X	X	X	everyone	All chat
/:id/cha t	PUT	X	Sender_n ame, sender_i d, message	X	everyone	

/investors:

endpoint	method	Request header	Request body	Request query	access	response
/:id?cha nnels	GET	Authoriz ation	X	Investor id	investor	All investor channels

/partners:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	X	X	X	everyone	All partners

/admin/events:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	Authoriz ation	X	X	Admin	Events
/:id	GET	Authoriz ation	X	Event id	Admin	Events
/	POST	Authoriz ation	Events details	X	Admin	
/:id	PUT	Authoriz ation	Events details	Event ID	Admin	

DOCUMENTATION – JEB Incubator

/:id	DELETE	Authoriz ation	X	Event ID	Admin	
------	--------	-------------------	---	----------	-------	--

/admin/investisor:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	Authoriz ation	X	X	Admin	Investisor s
/:id	GET	Authoriz ation	X	Investis or id	Admin	Investisor
/	POST	Authoriz ation	Investis or details	X	Admin	
/:id	PUT	Authoriz ation	Investis or details	Investis or ID	Admin	
/:id	DELETE	Authoriz ation	X	Investis or ID	Admin	

/admin/startup:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	Authoriz ation	X	X	Admin	Startups
/:id	GET	Authoriz ation	X	Startup id	Admin	Startup
/	POST	Authoriz ation	Startup details	X	Admin	
/:id	PUT	Authoriz ation	Startup details	Startup ID	Admin	
/:id	DELETE	Authoriz ation	X	Startup ID	Admin	

/admin/news:

DOCUMENTATION – JEB Incubator

endpoint	method	Request header	Request body	Request query	access	response
/	GET	Authorization	X	X	Admin	News
/:id	GET	Authorization	X	New id	Admin	New
/	POST	Authorization	New details	X	Admin	
/:id	PUT	Authorization	New details	New ID	Admin	
/:id	DELETE	Authorization	X	New ID	Admin	

/admin/partner:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	Authorization	X	X	Admin	Partners
/:id	GET	Authorization	X	Partner id	Admin	Partner
/	POST	Authorization	Partner details	X	Admin	
/:id	PUT	Authorization	Partner details	Partner ID	Admin	
/:id	DELETE	Authorization	X	Partner ID	Admin	

/admin/users:

endpoint	method	Request header	Request body	Request query	access	response
/	GET	Authorization	X	X	Admin	users
/:id	GET	Authorization	X	User id	Admin	User

DOCUMENTATION – JEB Incubator

/	POST	Authoriz ation	User details	X	Admin	
/:id	PUT	Authoriz ation	User details	User ID	Admin	
/:id	DELETE	Authoriz ation	X	User ID	Admin	

Contributors

- Quentin H.
- Emma H.
- Nicolas S.
- Lucas H.
- Loïc R.