COGNISCRUIT

Team Yoshi

Professor: *Mr. Pratik R Chaudhari*

Team members:
Raashil Aadhyanth
Jonathan Guan
Sujith Gowdru Prabhu Venkatesh
Ross Carvalho
Reshma Palasamudram Ramakumar
Hui Zhang
Arun Chowdary
Mahalakshmi Nagineni

Introduction

- Cogniscruit is an AI-powered platform that automatically generates personalized interview questions for recruiters.
- Leveraging advanced AI to analyze candidate profiles, resumes, and job descriptions.
- Cogniscruit generates tailored interview questions that save recruiters time and ensure consistency.

Target Audience

Recruiters & Hiring Managers

Interviewers

Job Seekers (for practice)

Students (for interview preparation)

Tech-Stack

NEXT.JS

Flask
web development,
one drop at a time

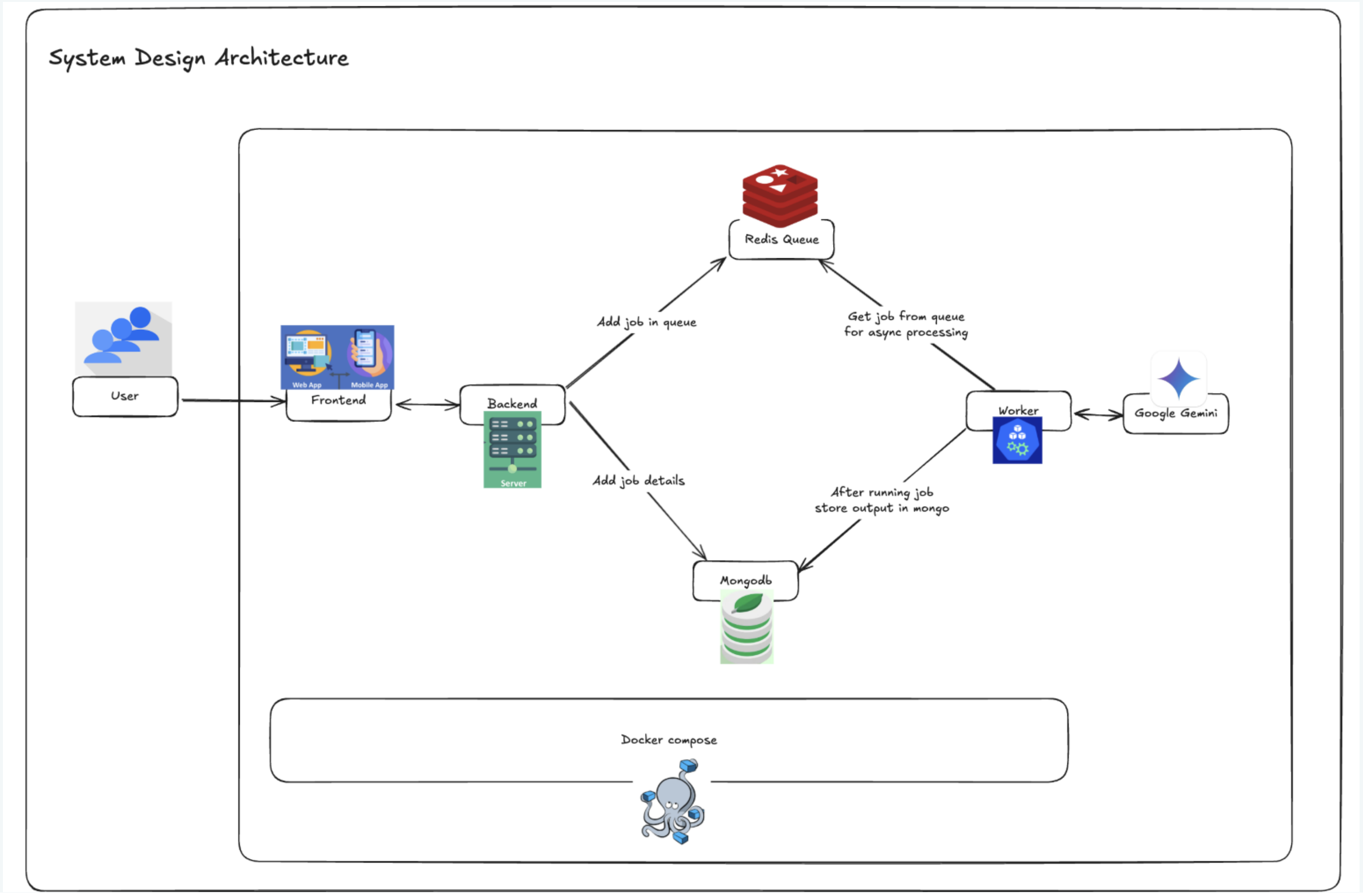
redis

mongoDB

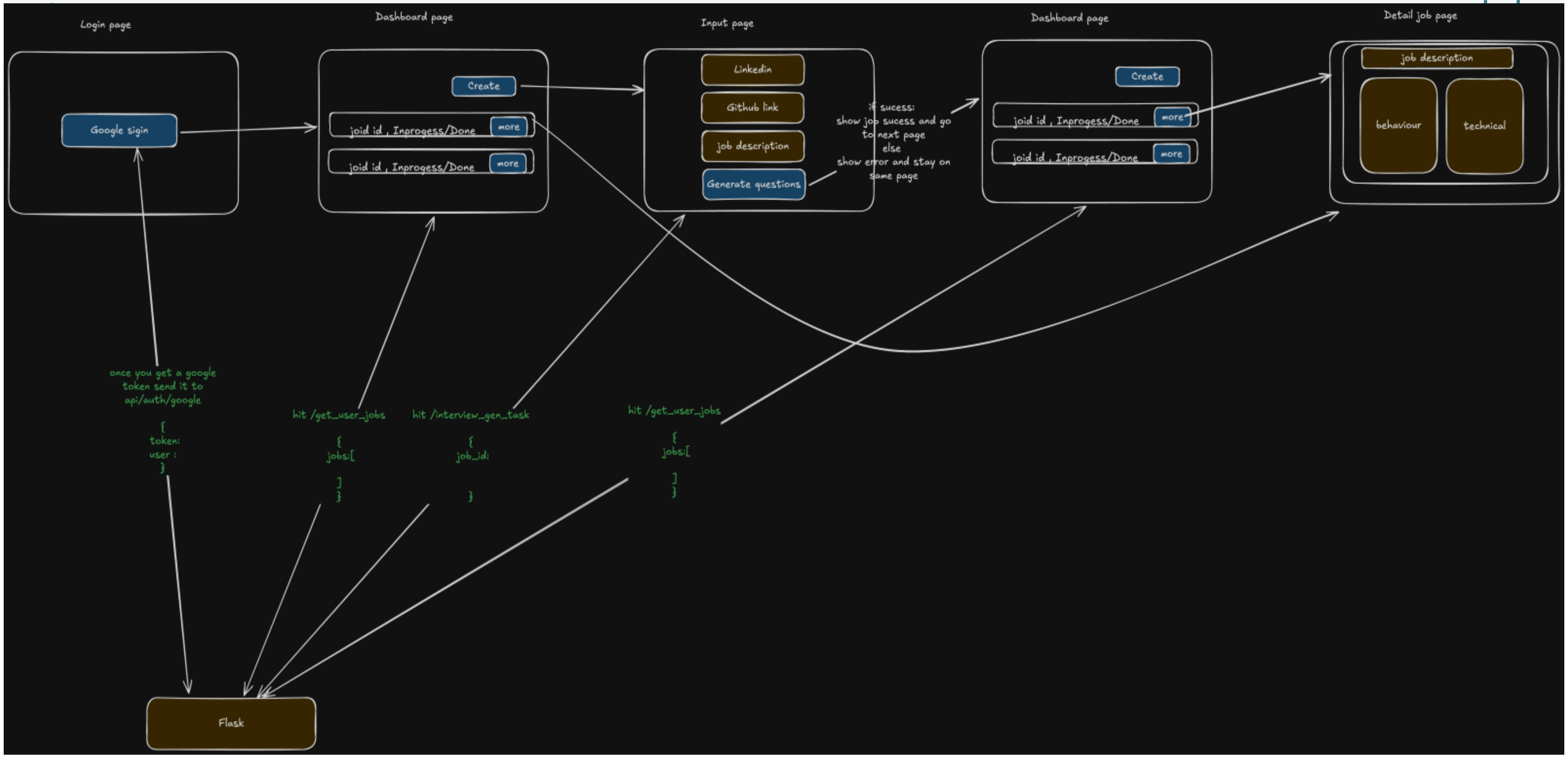
docker
Compose

docker

System Design



UI/UX Mocks


Login page
Google sigin
Dashboard page
Create
joid id , Inprogess/Done
more
joid id , Inprogess/Done
more
Input page
Linkedin
Github link
job description
Generate questions
if sucess:
show job sucess and go
to next page
else
show error and stay on
same page
Dashboard page
Create
joid id , Inprogess/Done
more
joid id , Inprogess/Done
more
Detail job page
job description
behaviour
technical
once you get a google
token send it to
api/auth/google
{
token:
user :
}
hit /get_user_jobs
{
jobs:[
]
}
hit /interview_gen_task
{
job_id:
}
hit /get_user_jobs
{
jobs:[
]
}
Flask

Internal APIs

- **POST /interview_gen_task :** Initiates interview content generation.
  - Input: Job description, GitHub username, LinkedIn URL.
  - Output: Job ID.
- **POST /get_user_jobs :** Retrieves a list of job-related tasks for the authenticated user.
  - Input: None (uses Bearer Token).
  - Output: List of jobs with details (description, links, status, generated questions).
- **GET /api/user/profile :** Fetches profile information of the authenticated user.
  - Input: None (uses Bearer Token).
  - Output: User details (ID, email, name, picture, creation date).
- **POST /api/auth/google :** Provides JWT token for accessing backend services via Google Sign-In.
  - Input: Google auth token.
  - Output: JWT token and user details.

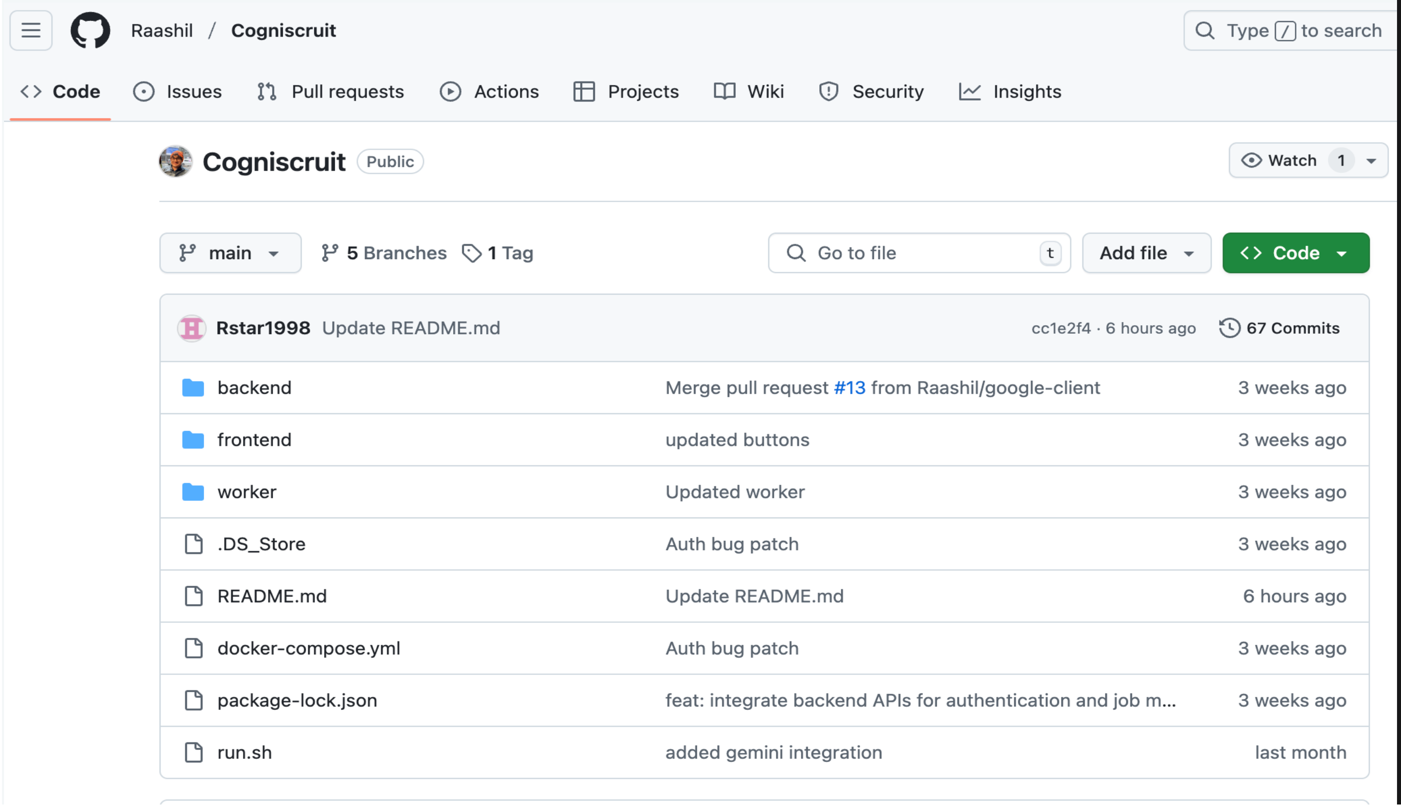
External APIs

- **Google Gemini API:** For AI-powered question generation.
- **GitHub API:** To fetch user repository data (via `github_service.py`).
- **LinkedIn API/Scraping:** To fetch user profile data (via `linkedin_service.py`).

Sprint Achievements

- **Sprint 1:** Established project foundation: Defined project scope and initial system architecture
- **Sprint 2:** Developed individual modules including backend, frontend, worker, databases and LLM.
- **Sprint 3 :** Integrated all modules into a microservice architecture and orchestrated it on Docker Compose
- **Sprint 4 :** Completed product and prepared for release: Conducted testing, fixed bugs, finalized documentation, and configured deployment.

Code Highlights



Raashil / Cogniscruit

Type / to search

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights

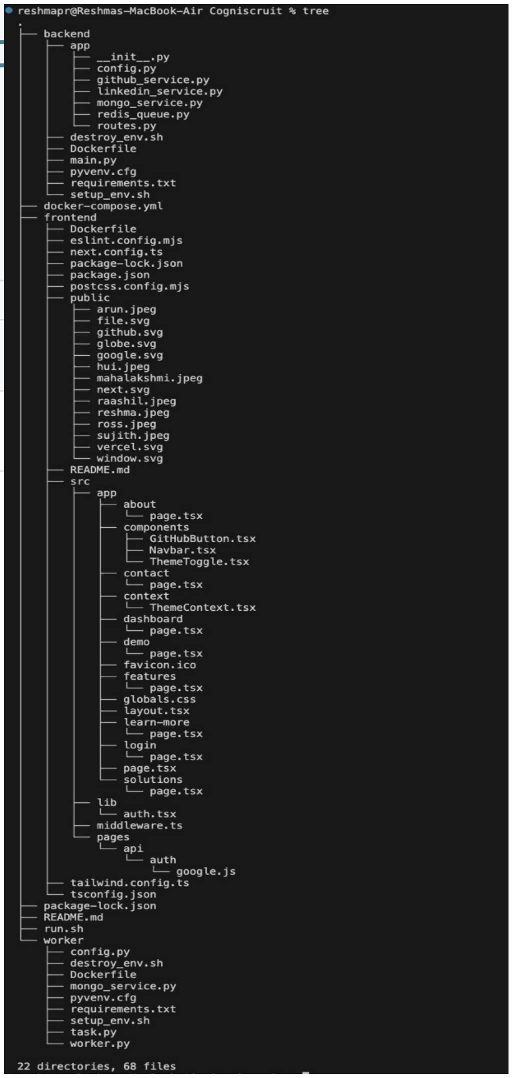
Cogniscruit Public

Watch 1

main · 5 Branches · 1 Tag · Go to file · Add file · Code

Rstar1998 Update README.md — cc1e2f4 · 6 hours ago — 67 Commits

backend	Merge pull request #13 from Raashil/google-client	3 weeks ago
frontend	updated buttons	3 weeks ago
worker	Updated worker	3 weeks ago
.DS_Store	Auth bug patch	3 weeks ago
README.md	Update README.md	6 hours ago
docker-compose.yml	Auth bug patch	3 weeks ago
package-lock.json	feat: integrate backend APIs for authentication and job m...	3 weeks ago
run.sh	added gemini integration	last month



```
reshmapr@Reshmas-MacBook-Air Cogniscruit % tree
.
├── backend
│   ├── app
│   │   ├── __init__.py
│   │   ├── config.py
│   │   ├── github_service.py
│   │   ├── linkedin_service.py
│   │   ├── mongo_service.py
│   │   ├── redis_queue.py
│   │   └── routes.py
│   ├── destroy_env.sh
│   ├── Dockerfile
│   ├── main.py
│   ├── pyvenv.cfg
│   ├── requirements.txt
│   └── setup_env.sh
├── docker-compose.yml
├── frontend
│   ├── Dockerfile
│   ├── eslint.config.mjs
│   ├── next.config.ts
│   ├── package-lock.json
│   ├── package.json
│   ├── postcss.config.mjs
│   ├── public
│   │   ├── arun.jpeg
│   │   ├── file.svg
│   │   ├── github.svg
│   │   ├── globe.svg
│   │   ├── google.svg
│   │   ├── hui.jpeg
│   │   ├── mahalakshmi.jpeg
│   │   ├── next.svg
│   │   ├── raashil.jpeg
│   │   ├── reshma.jpeg
│   │   ├── ross.jpeg
│   │   ├── sujith.jpeg
│   │   ├── vercel.svg
│   │   └── window.svg
│   ├── README.md
│   ├── src
│   │   ├── app
│   │   │   ├── about
│   │   │   │   └── page.tsx
│   │   │   ├── components
│   │   │   │   ├── GitHubButton.tsx
│   │   │   │   ├── Navbar.tsx
│   │   │   │   └── ThemeToggle.tsx
│   │   │   ├── contact
│   │   │   │   └── page.tsx
│   │   │   ├── context
│   │   │   │   └── ThemeContext.tsx
│   │   │   ├── dashboard
│   │   │   │   └── page.tsx
│   │   │   ├── demo
│   │   │   │   └── page.tsx
│   │   │   ├── favicon.ico
│   │   │   ├── features
│   │   │   │   └── page.tsx
│   │   │   ├── globals.css
│   │   │   ├── layout.tsx
│   │   │   ├── learn-more
│   │   │   │   └── page.tsx
│   │   │   ├── login
│   │   │   │   └── page.tsx
│   │   │   ├── page.tsx
│   │   │   └── solutions
│   │   │       └── page.tsx
│   │   ├── lib
│   │   │   └── auth.tsx
│   │   ├── middleware.ts
│   │   └── pages
│   │       └── api
│   │           └── auth
│   │               └── google.js
│   ├── tailwind.config.ts
│   └── tsconfig.json
├── package-lock.json
├── README.md
├── run.sh
└── worker
    ├── config.py
    ├── destroy_env.sh
    ├── Dockerfile
    ├── mongo_service.py
    ├── pyvenv.cfg
    ├── requirements.txt
    ├── setup_env.sh
    ├── task.py
    └── worker.py

22 directories, 68 files
```

Testing Overview

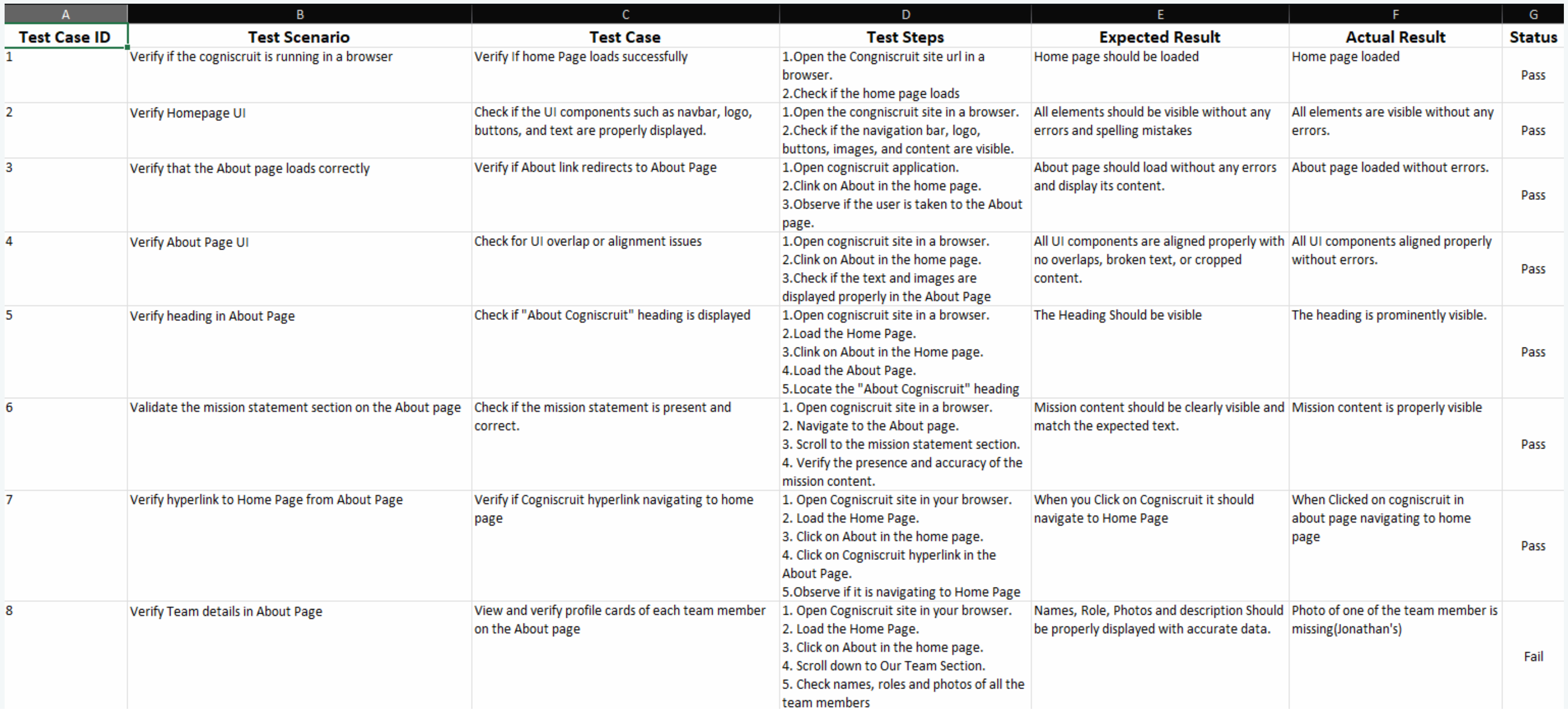
Objective:
Ensure end-to-end functionality, UI/UX consistency, and AI question relevance across the Cogniscruit application — from landing pages to the AI interview question generator and analytics dashboard.

Types of Testing Performed:
- Functional Testing
- Form Validation Testing
- Integration Testing
- AI Output Testing
- Responsiveness Testing

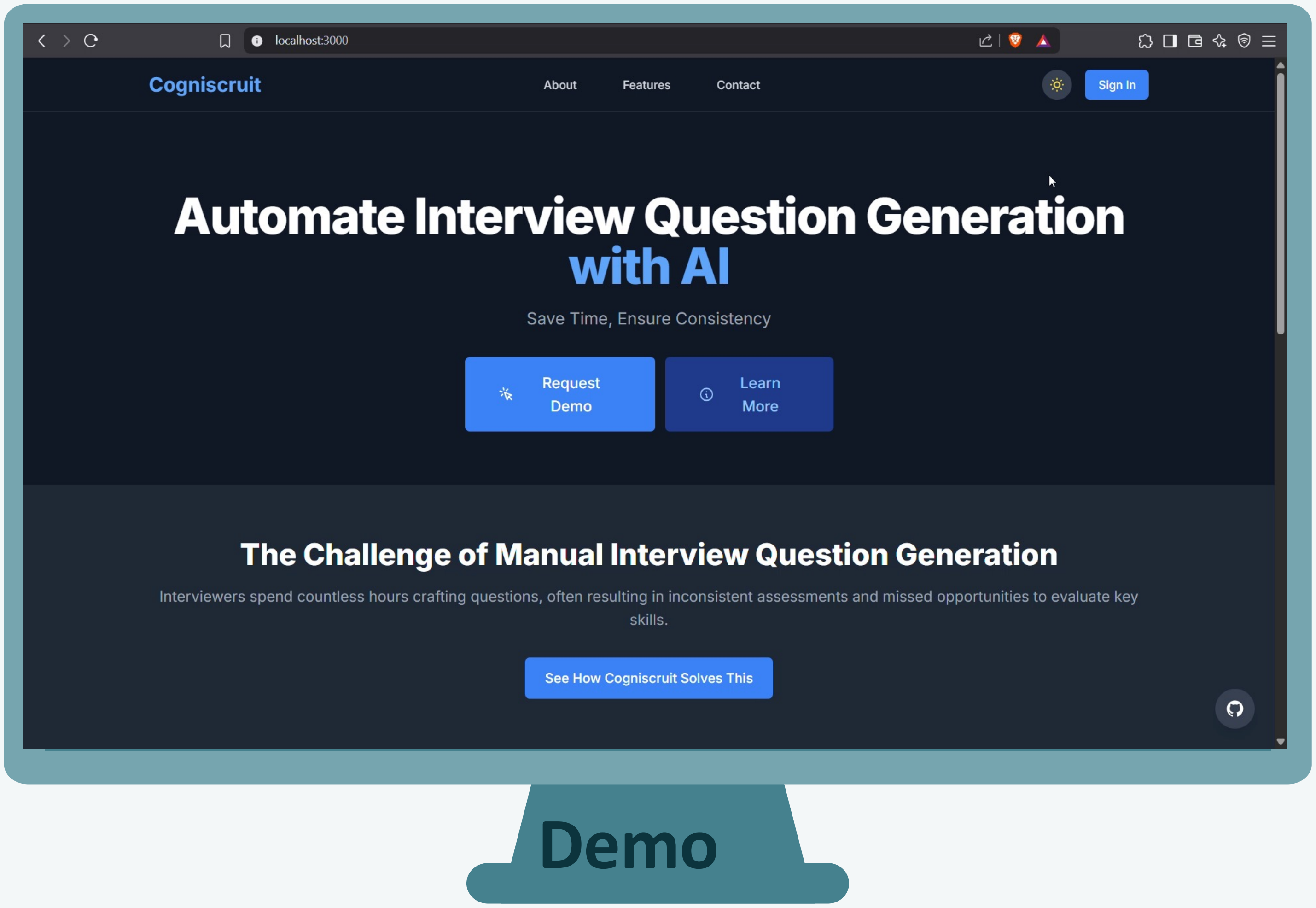
Testing Scope Covered:
- Home / About / Features / Contact / Learn More / Demo Pages
- Dashboard / Question Generator / Analytics
- Google Login / Form Validations / Edge Cases

Test Case Documentation:

- Maintained Word & Excel documentation for testcases.
- Structure included: Scenario, Test Case, Steps, Expected vs Actual, Status.



Test Case ID	Test Scenario	Test Case	Test Steps	Expected Result	Actual Result	Status
1	Verify if the cogniscruit is running in a browser	Verify If home Page loads successfully	1.Open the Congniscruit site url in a browser. 2.Check if the home page loads	Home page should be loaded	Home page loaded	Pass
2	Verify Homepage UI	Check if the UI components such as navbar, logo, buttons, and text are properly displayed.	1.Open the congniscruit site in a browser. 2.Check if the navigation bar, logo, buttons, images, and content are visible.	All elements should be visible without any errors and spelling mistakes	All elements are visible without any errors.	Pass
3	Verify that the About page loads correctly	Verify if About link redirects to About Page	1.Open cogniscruit application. 2.Clink on About in the home page. 3.Observe if the user is taken to the About page.	About page should load without any errors and display its content.	About page loaded without errors.	Pass
4	Verify About Page UI	Check for UI overlap or alignment issues	1.Open cogniscruit site in a browser. 2.Clink on About in the home page. 3.Check if the text and images are displayed properly in the About Page	All UI components are aligned properly with no overlaps, broken text, or cropped content.	All UI components aligned properly without errors.	Pass
5	Verify heading in About Page	Check if "About Cogniscruit" heading is displayed	1.Open cogniscruit site in a browser. 2.Load the Home Page. 3.Clink on About in the Home page. 4.Load the About Page. 5.Locate the "About Cogniscruit" heading	The Heading Should be visible	The heading is prominently visible.	Pass
6	Validate the mission statement section on the About page	Check if the mission statement is present and correct.	1. Open cogniscruit site in a browser. 2. Navigate to the About page. 3. Scroll to the mission statement section. 4. Verify the presence and accuracy of the mission content.	Mission content should be clearly visible and match the expected text.	Mission content is properly visible	Pass
7	Verify hyperlink to Home Page from About Page	Verify if Cogniscruit hyperlink navigating to home page	1. Open Cogniscruit site in your browser. 2. Load the Home Page. 3. Click on About in the home page. 4. Click on Cogniscruit hyperlink in the About Page. 5.Observe if it is navigating to Home Page	When you Click on Cogniscruit it should navigate to Home Page	When Clicked on cogniscruit in about page navigating to home page	Pass
8	Verify Team details in About Page	View and verify profile cards of each team member on the About page	1. Open Cogniscruit site in your browser. 2. Load the Home Page. 3. Click on About in the home page. 4. Scroll down to Our Team Section. 5. Check names, roles and photos of all the team members	Names, Role, Photos and description Should be properly displayed with accurate data.	Photo of one of the team member is missing(Jonathan's)	Fail


localhost:3000

Cogniscruit

About Features Contact

Sign In

Automate Interview Question Generation with AI

Save Time, Ensure Consistency

Request Demo

Learn More

The Challenge of Manual Interview Question Generation

Interviewers spend countless hours crafting questions, often resulting in inconsistent assessments and missed opportunities to evaluate key skills.

See How Cogniscruit Solves This

Demo

Future Updates & Additional Features

- Advanced analytics for question effectiveness.
- Ability to edit/customize generated questions.
- Direct integration with Applicant Tracking Systems (ATS).
- Support for more data sources (e.g., personal websites, publications).
- Different question difficulty levels.

Team Contribution

Raashil Aadhyanth(Team Lead)

- **Role**: Frontend Developer, Scrum Master
- Proposed and established a modern frontend stack with Next.js (TypeScript), organizing the repository using the latest app directory structure.
- Led the team as Scrum Master, overseeing agile workflows and progress tracking with Jira.
- Built a responsive home page with GitHub integration, dynamic navigation, theme toggle (light/dark mode), and an "About" section using Tailwind CSS and React Icons.
- Developed and optimized sign-in and dashboard pages, implementing secure authentication with NextAuth.js
- Integrated backend APIs with Axios for document upload, file processing, and question generation features.
- Ensured code quality, type safety, and scalability with TypeScript, ESLint, Docker support, and environment variable management.

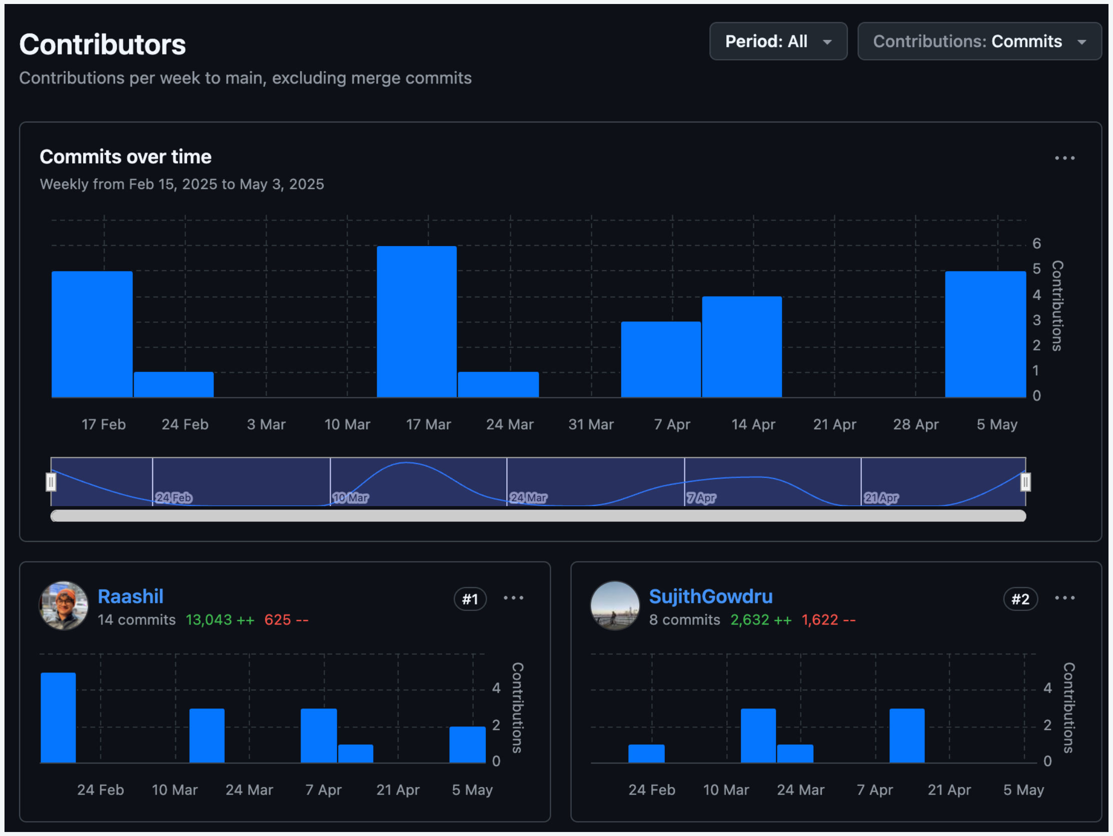
Sujith

- **Role** : Frontend Developer, UI/UX Engineer

- Built core frontend (signup, dashboard) using Next.js and TypeScript.
- Integrated Google authentication sign-in via React-Oauth.
- Connected dashboard uploads to backend APIs for question generation.
- Collaborated with backend and testing teams to resolve key issues.
- Improved performance and implemented single-auth flow for final release.

Frontend – GitHub Stats

Commits



Contributors

Contributions per week to main, excluding merge commits

Period: All

Contributions: Commits

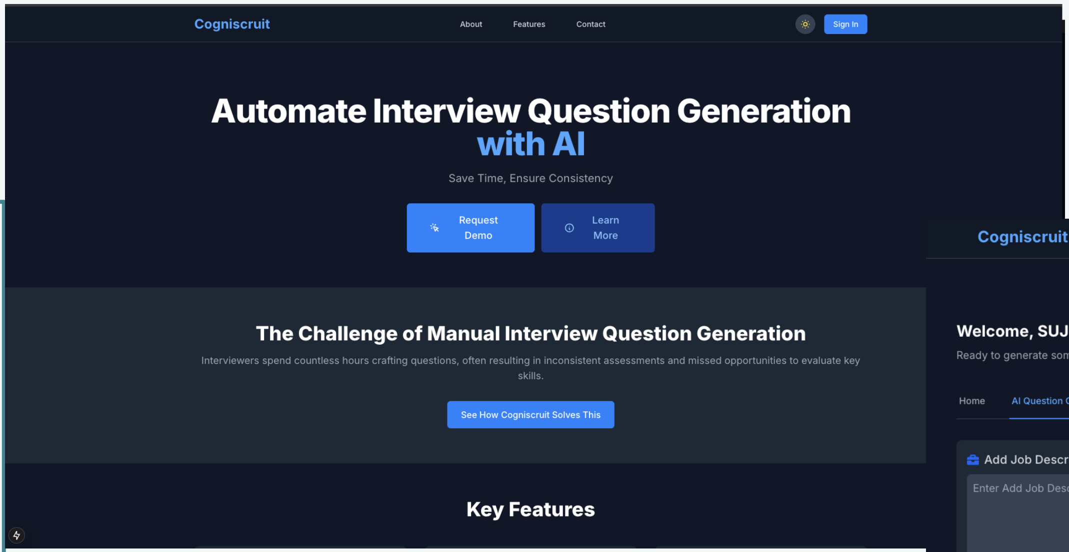
Commits over time

Weekly from Feb 15, 2025 to May 3, 2025

Raashil #1
14 commits 13,043 ++ 625 --

SujithGowdru #2
8 commits 2,632 ++ 1,622 --

Frontend Team -



Cogniscruit About Features Contact Sign In

Automate Interview Question Generation with AI

Save Time, Ensure Consistency

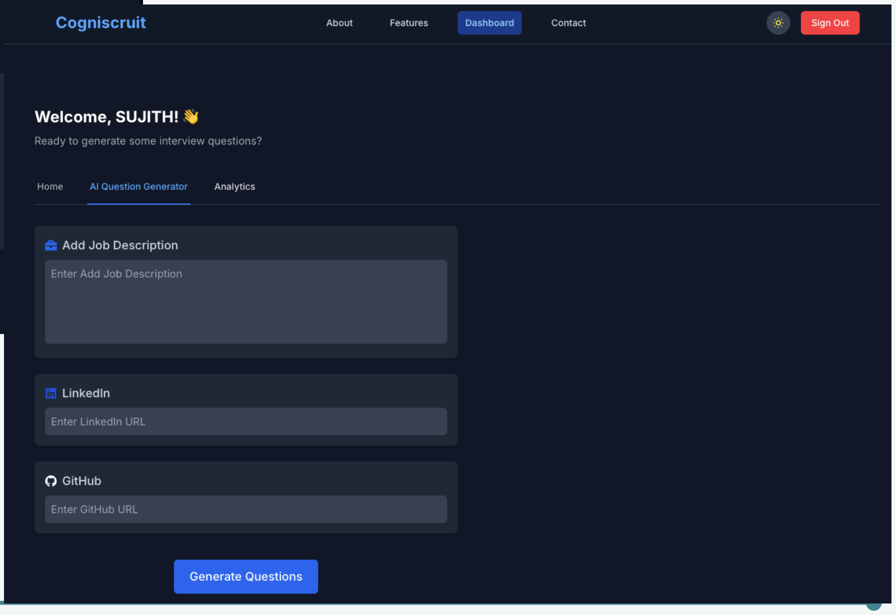
Request Demo | Learn More

The Challenge of Manual Interview Question Generation

Interviewers spend countless hours crafting questions, often resulting in inconsistent assessments and missed opportunities to evaluate key skills.

See How Cogniscruit Solves This

Key Features



Cogniscruit About Features Dashboard Contact Sign Out

Welcome, SUJITH! 👋

Ready to generate some interview questions?

Home | AI Question Generator | Analytics

Add Job Description: Enter Add Job Description

LinkedIn: Enter LinkedIn URL

GitHub: Enter GitHub URL

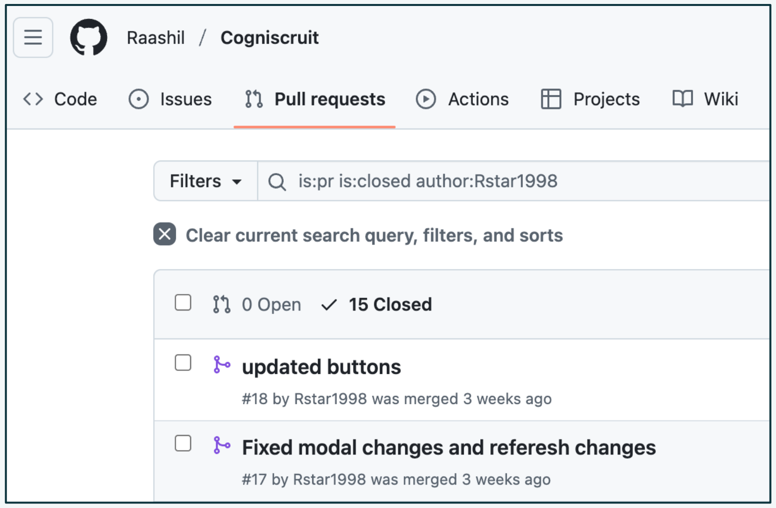
Generate Questions

Ross Carvalho

- **Role** : Solution Architect , Backend Developer
- Implemented
  - Backend Service (Python)
  - Worker Service (Python)
- Integrated Redis and MongoDB with Backend and Worker
- Dockerize all components
- Orchestrated all Docker microservices on Docker Compose

Ross Carvalho – GitHub Stats

Pull request closed



Raashil / Cogniscruit

Code · Issues · Pull requests · Actions · Projects · Wiki

Filters · is:pr is:closed author:Rstar1998

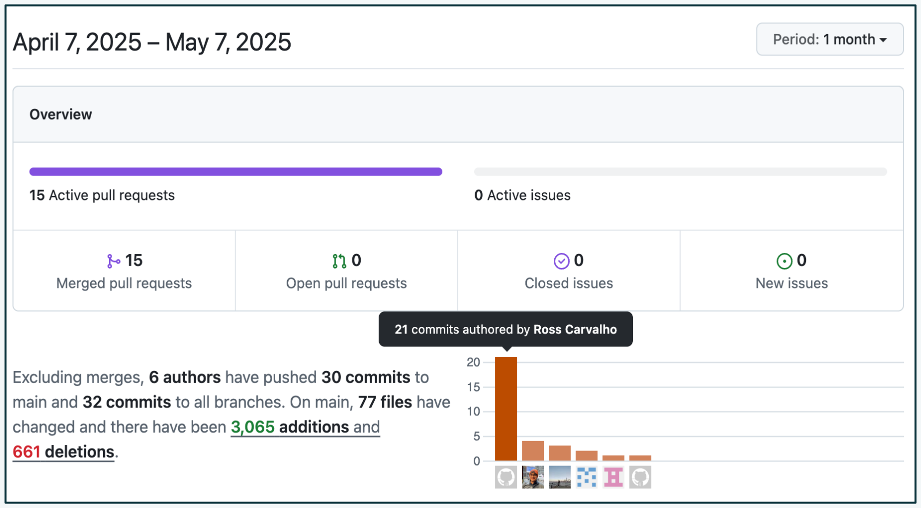
Clear current search query, filters, and sorts

0 Open ✓ 15 Closed

updated buttons
#18 by Rstar1998 was merged 3 weeks ago

Fixed modal changes and referesh changes
#17 by Rstar1998 was merged 3 weeks ago

Commits



April 7, 2025 – May 7, 2025

Period: 1 month

Overview

15 Active pull requests

0 Active issues

15	0	0	0
Merged pull requests	Open pull requests	Closed issues	New issues

21 commits authored by Ross Carvalho

Excluding merges, **6 authors** have pushed **30 commits** to main and **32 commits** to all branches. On main, **77 files** have changed and there have been **3,065 additions** and **661 deletions**.

Reshma

- **Role** : DevOps Engineer , Solution Architect

- Implemented
  - MongoDB
  - Confluent cloud.

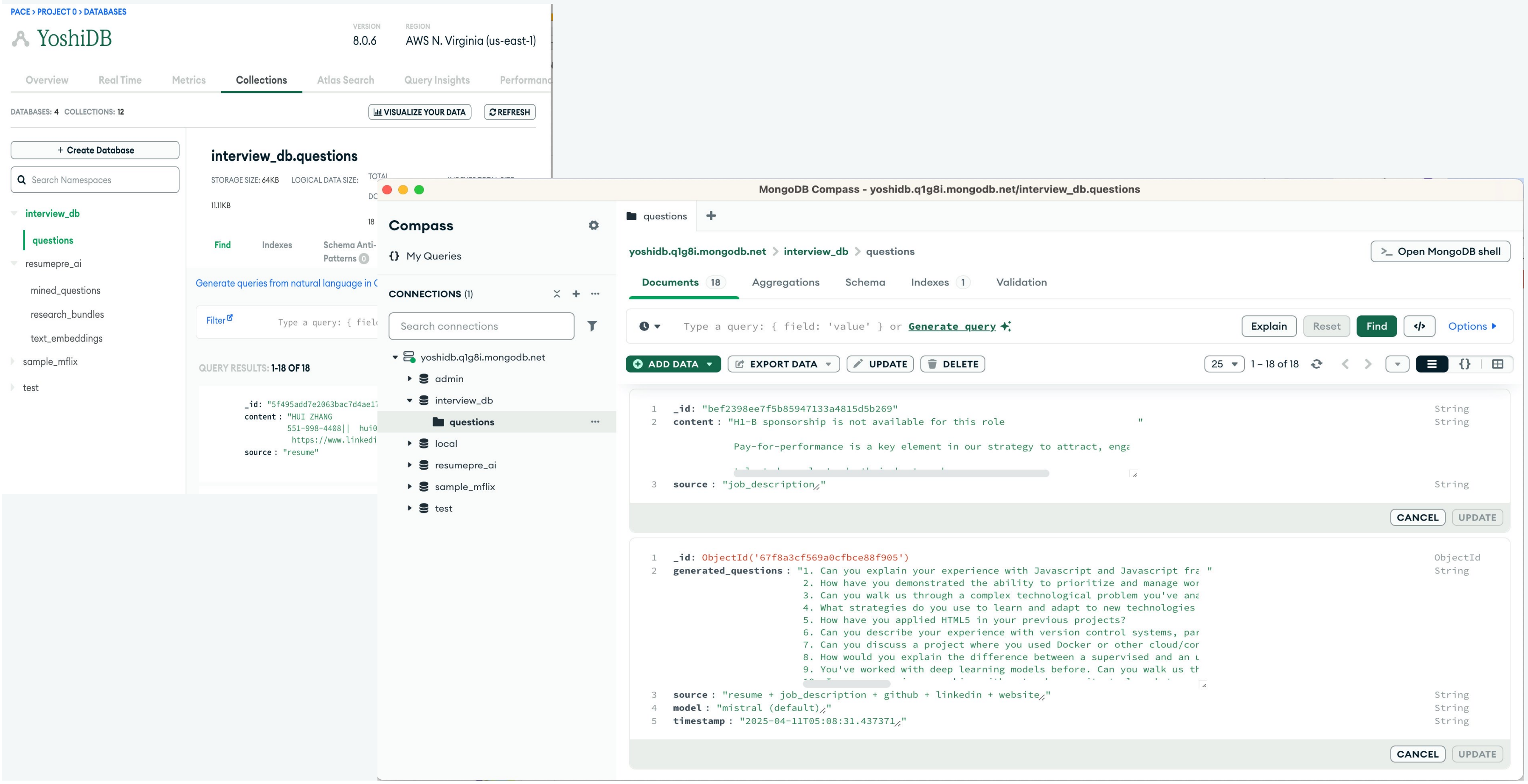
- Integrated MongoDB with AI and Confluent cloud.

- Performed POC on MongoDB, Confluent cloud, Apache Flink.

- Architected Docker images on Docker Compose.

- Worked on Kubernetes.

MongoDB



PACE > PROJECT 0 > DATABASES

YoshiDB — VERSION 8.0.6 — REGION AWS N. Virginia (us-east-1)

DATABASES: 4 COLLECTIONS: 12

interview_db.questions

STORAGE SIZE: 64KB LOGICAL DATA SIZE: 11.11KB

QUERY RESULTS: 1-18 OF 18

```
_id: "5f495add7e2063bac7d4ae17
content : "HUI ZHANG
          551-998-4408||  hui0
           https://www.linkedi
source : "resume"
```

MongoDB Compass - yoshidb.q1g8i.mongodb.net/interview_db.questions

CONNECTIONS (1)

- yoshidb.q1g8i.mongodb.net
  - admin
  - interview_db
    - questions
  - local
  - resumepre_ai
  - sample_mflix
  - test

yoshidb.q1g8i.mongodb.net > interview_db > questions

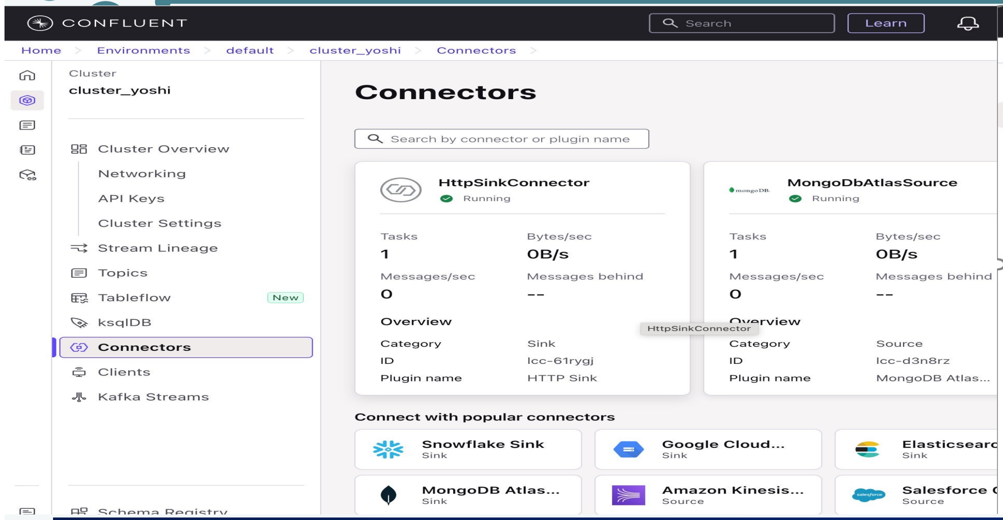
Documents 18 · Aggregations · Schema · Indexes 1 · Validation

25 · 1 – 18 of 18

```
_id: "bef2398ee7f5b85947133a4815d5b269"                                           String
content : "H1-B sponsorship is not available for this role                "       String

          Pay-for-performance is a key element in our strategy to attract, enga
source : "job_description"                                                        String
```

```
_id: ObjectId('67f8a3cf569a0cfbce88f905')                                         ObjectId
generated_questions : "1. Can you explain your experience with Javascript and Javascript fra "   String
                       2. How have you demonstrated the ability to prioritize and manage wor
                       3. Can you walk us through a complex technological problem you've ana
                       4. What strategies do you use to learn and adapt to new technologies
                       5. How have you applied HTML5 in your previous projects?
                       6. Can you describe your experience with version control systems, par
                       7. Can you discuss a project where you used Docker or other cloud/cor
                       8. How would you explain the difference between a supervised and an u
                       9. You've worked with deep learning models before. Can you walk us th
source : "resume + job_description + github + linkedin + website"                 String
model : "mistral (default)"                                                       String
timestamp : "2025-04-11T05:08:31.437371"                                          String
```



CONFLUENT — Search — Learn

Home > Environments > default > cluster_yoshi > Connectors >

Cluster
cluster_yoshi

- Cluster Overview
- Networking
- API Keys
- Cluster Settings
- Stream Lineage
- Topics
- Tableflow New
- ksqlDB
- Connectors
- Clients
- Kafka Streams
- Schema Registry

Connectors

Search by connector or plugin name

HttpSinkConnector — Running

Tasks	Bytes/sec
1	0B/s
Messages/sec	Messages behind
0	--

Overview

Category	Sink
ID	lcc-61rygj
Plugin name	HTTP Sink

MongoDbAtlasSource — Running

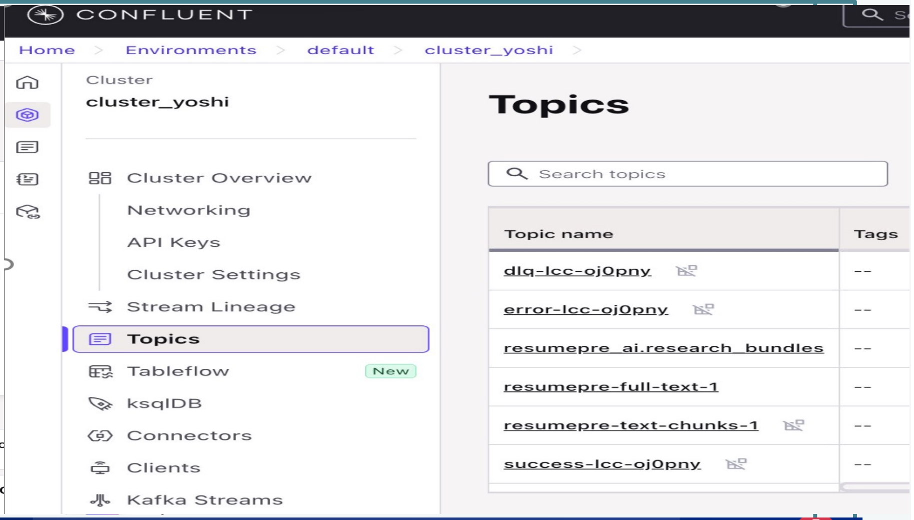
Tasks	Bytes/sec
1	0B/s
Messages/sec	Messages behind
0	--

Overview

Category	Source
ID	lcc-d3n8rz
Plugin name	MongoDB Atlas...

Connect with popular connectors

- Snowflake Sink — Sink
- Google Cloud... — Sink
- Elasticsearc — Sink
- MongoDB Atlas... — Sink
- Amazon Kinesis... — Source
- Salesforce C — Source



CONFLUENT

Home > Environments > default > cluster_yoshi >

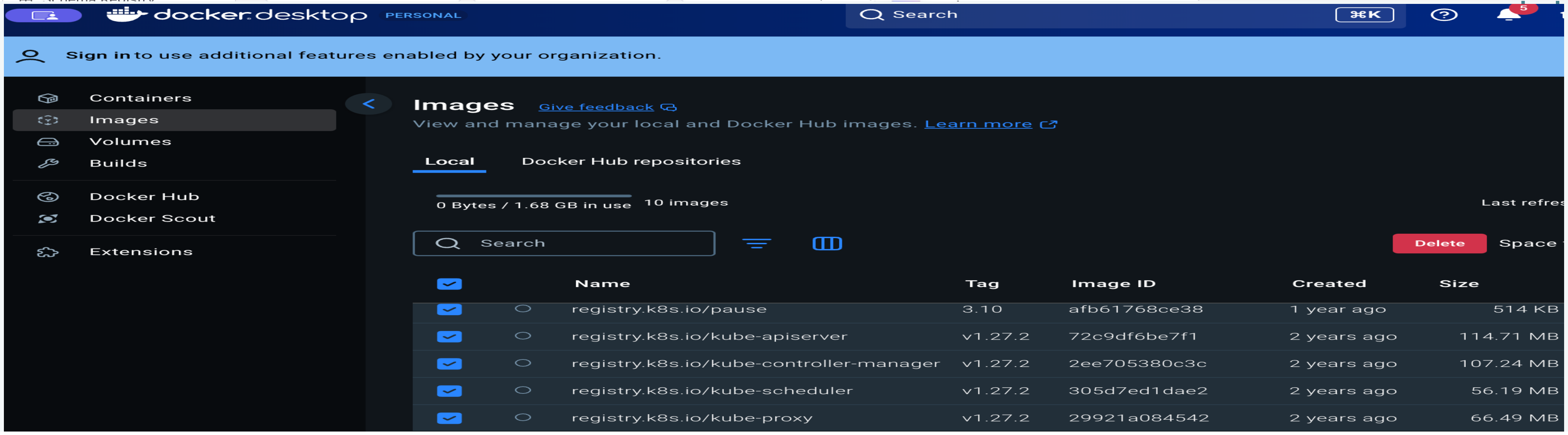
Cluster
cluster_yoshi

- Cluster Overview
- Networking
- API Keys
- Cluster Settings
- Stream Lineage
- Topics
- Tableflow New
- ksqlDB
- Connectors
- Clients
- Kafka Streams

Topics

Search topics

Topic name	Tags
dlq-lcc-oj0pny	--
error-lcc-oj0pny	--
resumepre_ai.research_bundles	--
resumepre-full-text-1	--
resumepre-text-chunks-1	--
success-lcc-oj0pny	--



docker desktop PERSONAL — Search ⌘K

Sign in to use additional features enabled by your organization.

- Containers
- Images
- Volumes
- Builds
- Docker Hub
- Docker Scout
- Extensions

Images

Give feedback

View and manage your local and Docker Hub images. Learn more

Local — Docker Hub repositories

0 Bytes / 1.68 GB in use — 10 images — Last refres

Search — Delete — Space

Name	Tag	Image ID	Created	Size
registry.k8s.io/pause	3.10	afb61768ce38	1 year ago	514 KB
registry.k8s.io/kube-apiserver	v1.27.2	72c9df6be7f1	2 years ago	114.71 MB
registry.k8s.io/kube-controller-manager	v1.27.2	2ee705380c3c	2 years ago	107.24 MB
registry.k8s.io/kube-scheduler	v1.27.2	305d7ed1dae2	2 years ago	56.19 MB
registry.k8s.io/kube-proxy	v1.27.2	29921a084542	2 years ago	66.49 MB

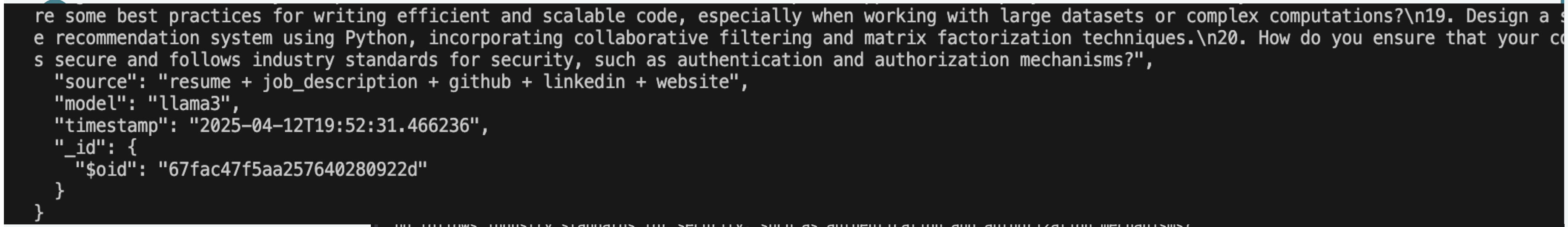
Johnathan

- Role : AI Engineer, LLM Architect
- Implemented
  - LangChain orchestration
  - Interactive follow-up system
- Designed personalized question generation pipeline using Mistral, LLaMA3 via Ollama
- Integrated MongoDB to manage session state and store generated questions
- Built RESTful API endpoints to enable seamless integration with frontend
- Led technical documentation, testing, and optimization for the AI module

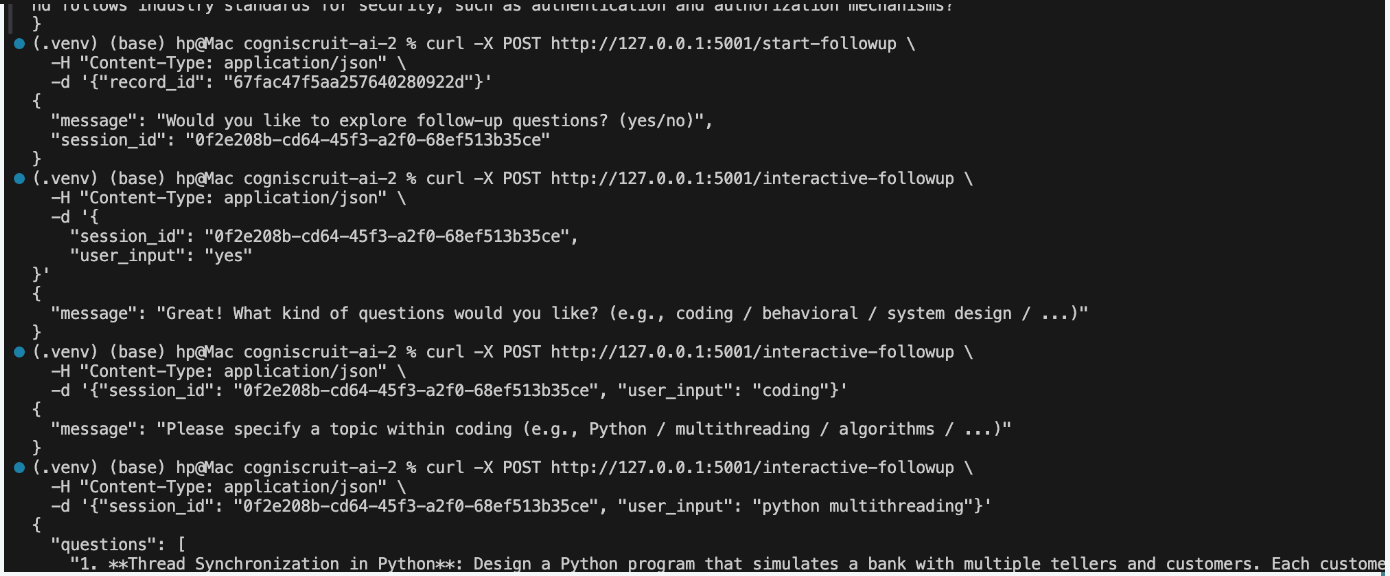
Hui

- **Role** : AI Engineer, NLP Specialist
- Implemented
  - Resume and job description parsing
  - Data enrichment agents
- Integrated API-based profile fetching for GitHub and LinkedIn data
- Developed semantic similarity matching with FAISS and embedding models
- Contributed to interactive system design and MongoDB integration
- Led system architecture design for flexibility and scalability

AI Team – local LLM model



```
re some best practices for writing efficient and scalable code, especially when working with large datasets or complex computations?\n19. Design a s
e recommendation system using Python, incorporating collaborative filtering and matrix factorization techniques.\n20. How do you ensure that your co
s secure and follows industry standards for security, such as authentication and authorization mechanisms?",
  "source": "resume + job_description + github + linkedin + website",
  "model": "llama3",
  "timestamp": "2025-04-12T19:52:31.466236",
  "_id": {
    "$oid": "67fac47f5aa257640280922d"
  }
}
```



```
nd follows industry standards for security, such as authentication and authorization mechanisms?"
}
(.venv) (base) hp@Mac cogniscruit-ai-2 % curl -X POST http://127.0.0.1:5001/start-followup \
  -H "Content-Type: application/json" \
  -d '{"record_id": "67fac47f5aa257640280922d"}'
{
  "message": "Would you like to explore follow-up questions? (yes/no)",
  "session_id": "0f2e208b-cd64-45f3-a2f0-68ef513b35ce"
}
(.venv) (base) hp@Mac cogniscruit-ai-2 % curl -X POST http://127.0.0.1:5001/interactive-followup \
  -H "Content-Type: application/json" \
  -d '{
    "session_id": "0f2e208b-cd64-45f3-a2f0-68ef513b35ce",
    "user_input": "yes"
}'
{
  "message": "Great! What kind of questions would you like? (e.g., coding / behavioral / system design / ...)"
}
(.venv) (base) hp@Mac cogniscruit-ai-2 % curl -X POST http://127.0.0.1:5001/interactive-followup \
  -H "Content-Type: application/json" \
  -d '{"session_id": "0f2e208b-cd64-45f3-a2f0-68ef513b35ce", "user_input": "coding"}'
{
  "message": "Please specify a topic within coding (e.g., Python / multithreading / algorithms / ...)"
}
(.venv) (base) hp@Mac cogniscruit-ai-2 % curl -X POST http://127.0.0.1:5001/interactive-followup \
  -H "Content-Type: application/json" \
  -d '{"session_id": "0f2e208b-cd64-45f3-a2f0-68ef513b35ce", "user_input": "python multithreading"}'
{
  "questions": [
    "1. **Thread Synchronization in Python**: Design a Python program that simulates a bank with multiple tellers and customers. Each custome
```

Arun Chowdary

- **Role** : Tester, Documenter

- Prepared and organized both Excel and Word files for test case tracking.
- Focused on integration testing, including navigation flow and post-login behavior.
- Verified dashboard functionality, job creation, and question generation module.
- Ensured responsiveness and alignment across screen sizes and browsers.
- Cross-checked question consistency across similar Job Descriptions (JDs).
- Supported issue tracking and updates via JIRA. And Contributed to project testing documentation/User Guide.

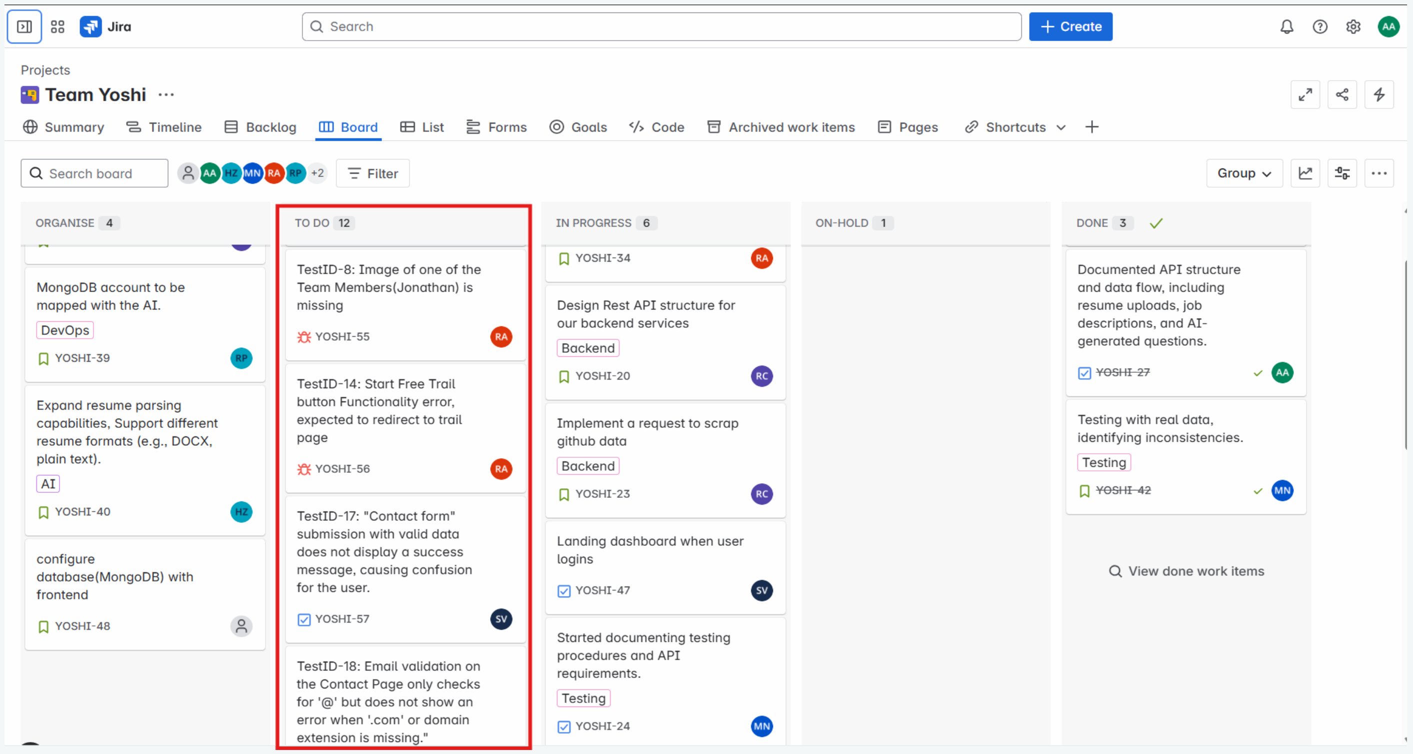
Mahalakshmi

- **Role** : Tester, Documenter

- Designed and executed UI and functional test cases.
- Verified AI-generated question relevance for uploaded Job Description, LinkedIn, and GitHub inputs.
- Ensured proper form validation, including email format, empty field handling, and URL correctness.
- Tested key user flows: Request Demo, Sign In, and Learn More buttons.
- Documented test cases
- Identified and reported bugs in forms and dashboard
- Contributed to the User guide, Testing in the Project Documentation.

Testing Team – Bug Tracking & JIRA Contributions

Raised bugs in Jira and coordinated fixes with the team



ORGANISE 4

- MongoDB account to be mapped with the AI. — DevOps — YOSHI-39 — RP
- Expand resume parsing capabilities, Support different resume formats (e.g., DOCX, plain text). — AI — YOSHI-40 — HZ
- configure database(MongoDB) with frontend — YOSHI-48

TO DO 12

- TestID-8: Image of one of the Team Members(Jonathan) is missing — YOSHI-55 — RA
- TestID-14: Start Free Trail button Functionality error, expected to redirect to trail page — YOSHI-56 — RA
- TestID-17: "Contact form" submission with valid data does not display a success message, causing confusion for the user. — YOSHI-57 — SV
- TestID-18: Email validation on the Contact Page only checks for '@' but does not show an error when '.com' or domain extension is missing."

IN PROGRESS 6

- YOSHI-34 — RA
- Design Rest API structure for our backend services — Backend — YOSHI-20 — RC
- Implement a request to scrap github data — Backend — YOSHI-23 — RC
- Landing dashboard when user logins — YOSHI-47 — SV
- Started documenting testing procedures and API requirements. — Testing — YOSHI-24 — MN

ON-HOLD 1

DONE 3

- Documented API structure and data flow, including resume uploads, job descriptions, and AI-generated questions. — ~~YOSHI-27~~ — AA
- Testing with real data, identifying inconsistencies. — Testing — ~~YOSHI-42~~ — MN

View done work items

Thank you