Skill’s Application : A Step-by-Step User Guide

Introduction:

Overview:

Introducing Skill's Application, the ultimate application designed specifically for software developers to guide them in advancing their skills. With Skill's Application, you can get your own roadmap or generate a personalized pdf based on the skillset you wish to learn which would act as a guide in your learning journey.

Key Benefits:

- Instead of manually crafting skill documents, the application automates the process, saving users significant time
- Users can easily update their skill tags and regenerate PDFs, ensuring their documents are always up-to-date with minimal effort.

Sequence:

Follow the following sequence if you are a new admin:

1. Getting Started
2. File Upload
3. Edit matrix
4. Load Matrix
5. Export Matrix

Additional functionalities:

1. Roadmap
2. Token based skill-matrix

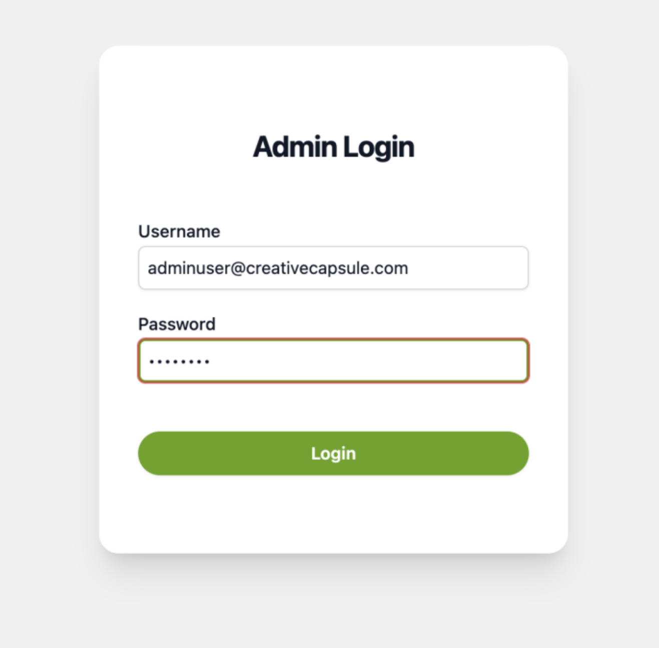
Functionalities of logged in user:

1. Skill Rating
2. Skill Matrix

Getting Started:

Login to the application:

Enter your username and password and login to the application.



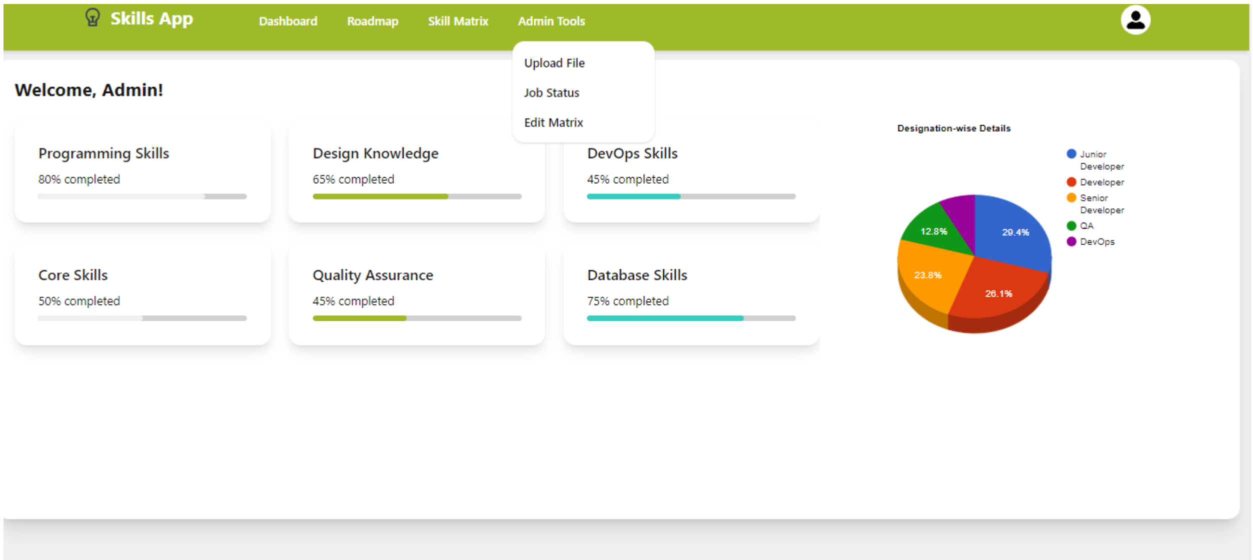
Username: adminuser@creativecapsule.com

Password: ••••••••

Features and Functions:

1. File Upload:

Navigate from the dashboard through Admin Tools >> Upload File



Welcome, Admin!

Programming Skills
80% completed

Design Knowledge
65% completed

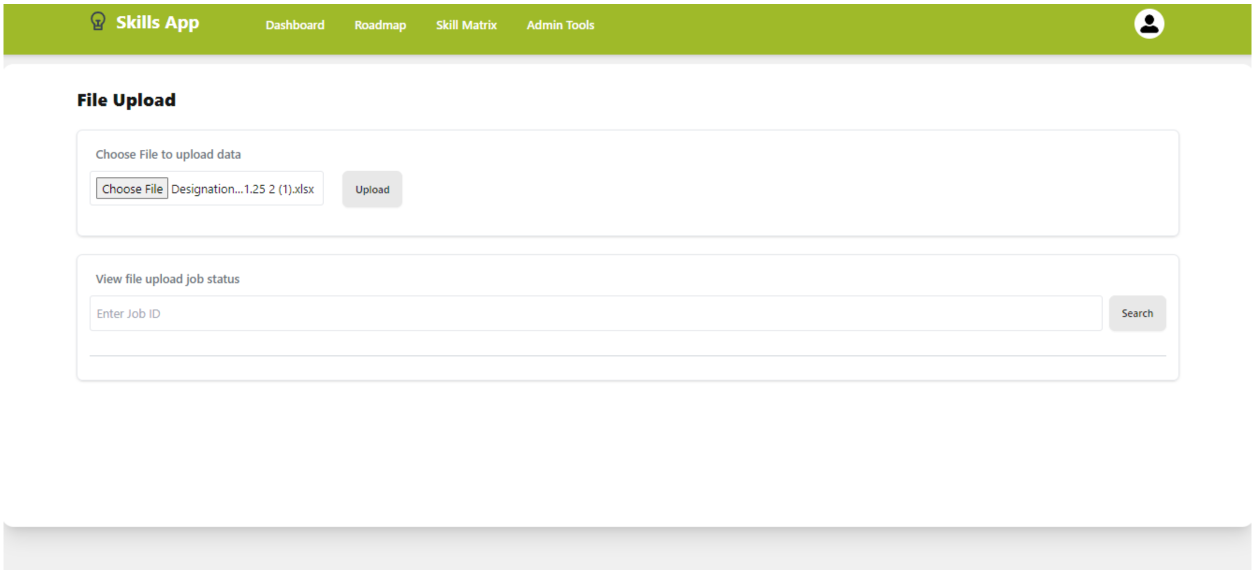
DevOps Skills
45% completed

Core Skills
50% completed

Quality Assurance
45% completed

Database Skills
75% completed

The admin user can pass an excel file with details of all the skills and their requirements of how to learn them which is then processed and is used from the database.



File Upload

Choose File to upload data

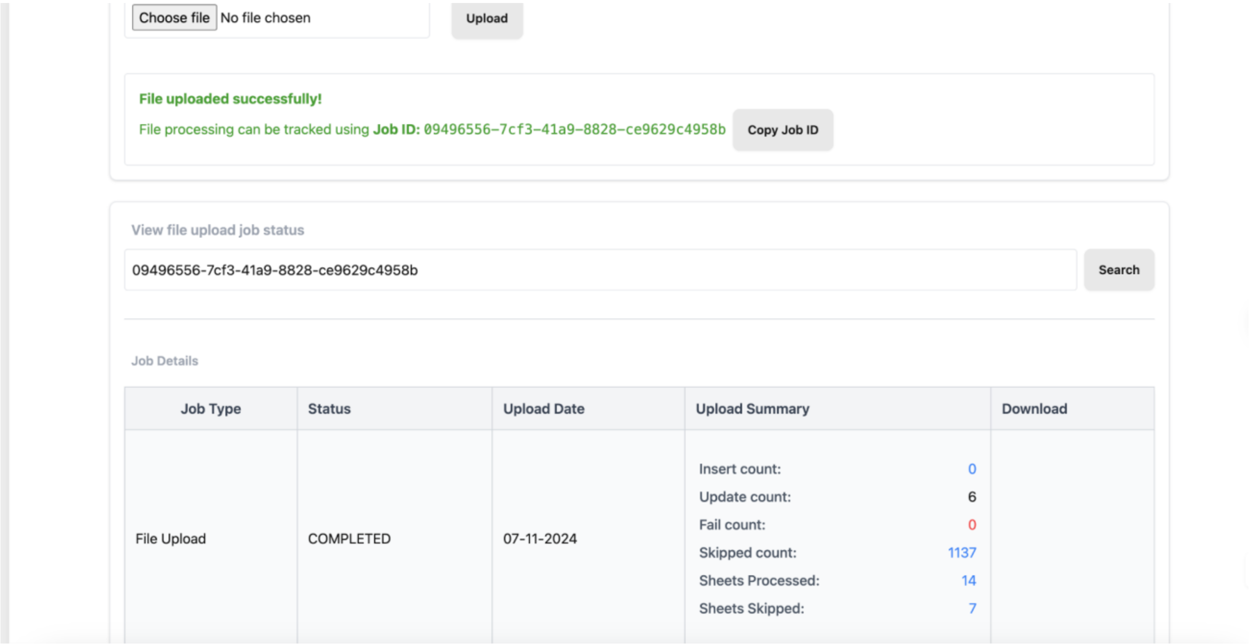
Designation...1.25 2 (1).xlsx

View file upload job status

Enter Job ID

2. Job Status:

The job status can be fetched using the job id which is displayed after uploading an excel file



No file chosen

File uploaded successfully!

File processing can be tracked using **Job ID:** `09496556-7cf3-41a9-8828-ce9629c4958b`

View file upload job status

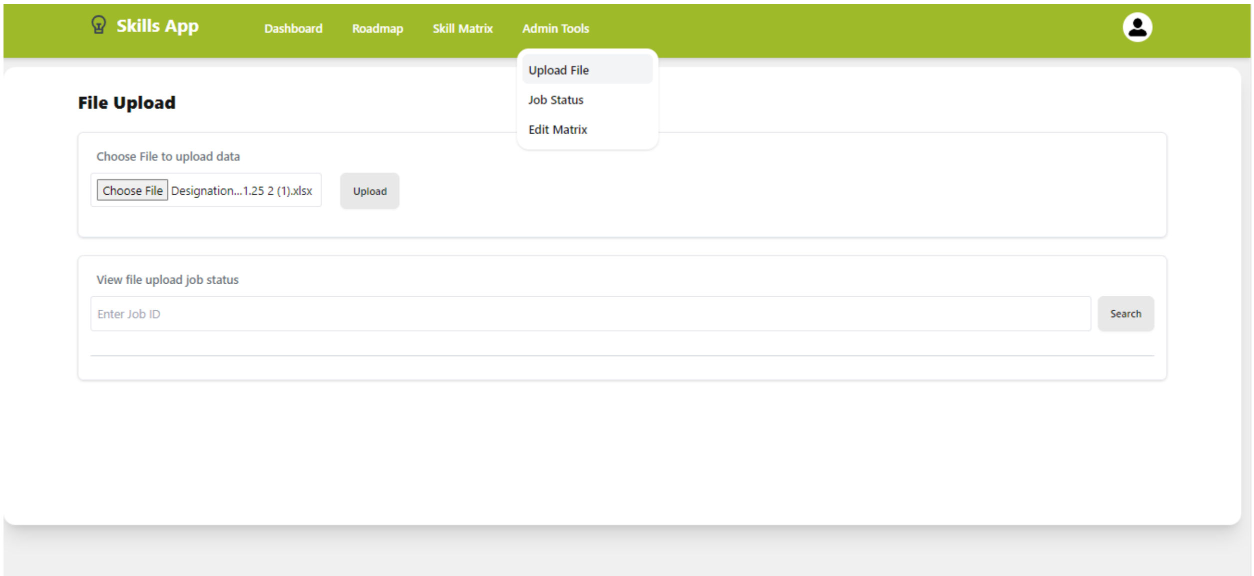
09496556-7cf3-41a9-8828-ce9629c4958b

Job Details

Job Type	Status	Upload Date	Upload Summary	Download
File Upload	COMPLETED	07-11-2024	Insert count: 0 Update count: 6 Fail count: 0 Skipped count: 1137 Sheets Processed: 14 Sheets Skipped: 7	

3. Job list:

Navigate through Admin Tools >> Job Status



File Upload

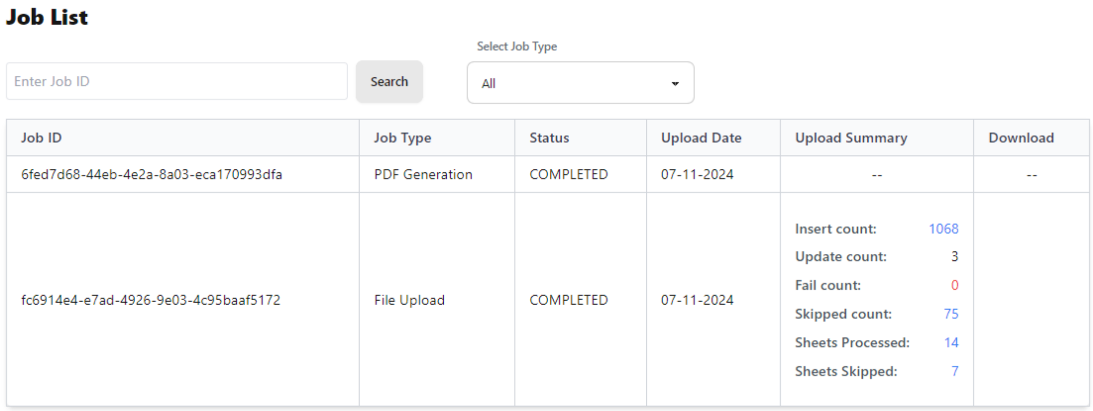
Choose File to upload data

Designation...1.25 2 (1).xlsx

View file upload job status

Enter Job ID

Job list of all the jobs can be seen and filtered based on job type or using its job Id



Job List

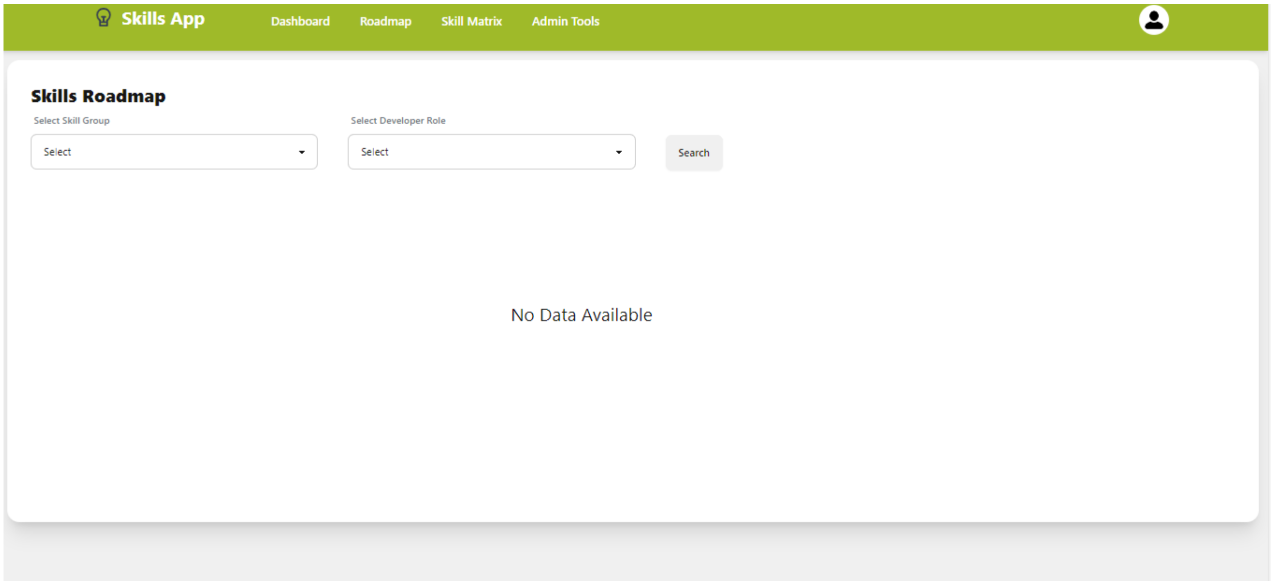
Enter Job ID

Select Job Type: All

Job ID	Job Type	Status	Upload Date	Upload Summary	Download
6fed7d68-44eb-4e2a-8a03-eca170993dfa	PDF Generation	COMPLETED	07-11-2024	--	--
fc6914e4-e7ad-4926-9e03-4c95baaf5172	File Upload	COMPLETED	07-11-2024	Insert count: 1068 Update count: 3 Fail count: 0 Skipped count: 75 Sheets Processed: 14 Sheets Skipped: 7	

4. Roadmap:

Navigate through Roadmap option on the navigation bar ;-



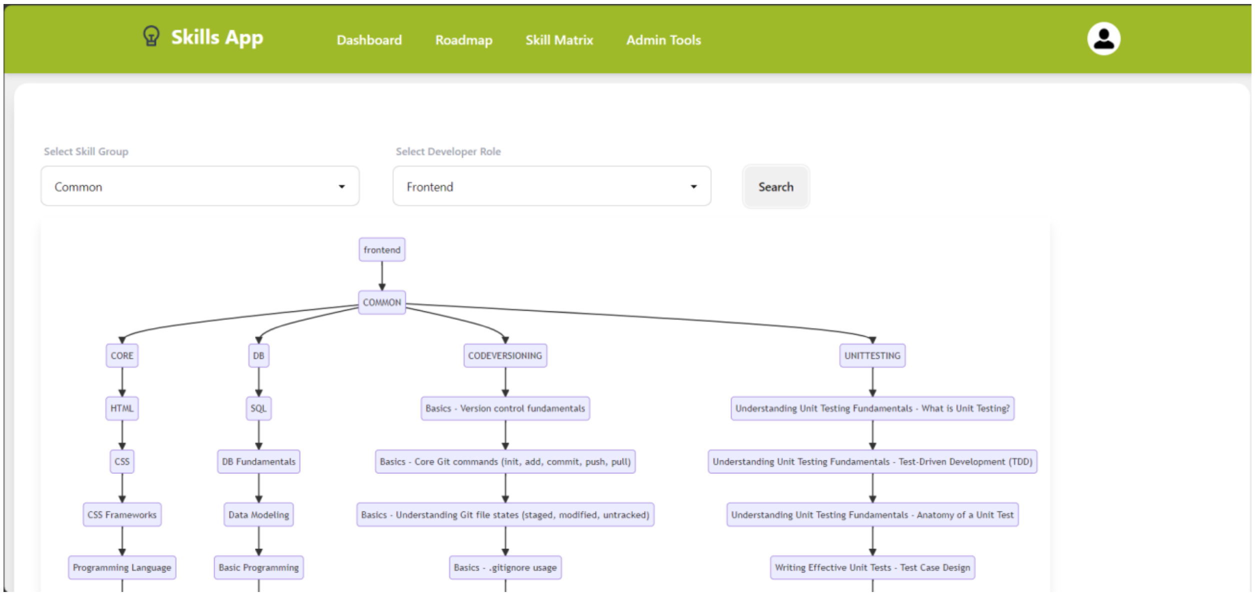
Skills Roadmap

Select Skill Group: Select

Select Developer Role: Select

No Data Available

Select desired frameworks and designation role and get the roadmap

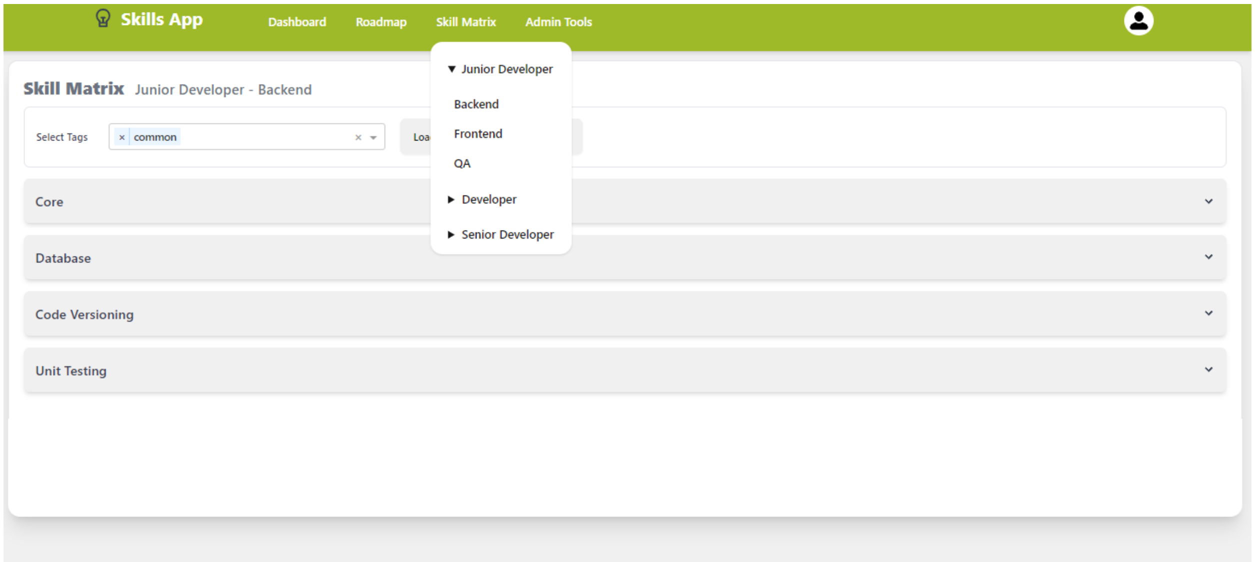

Skills App
Dashboard
Roadmap
Skill Matrix
Admin Tools
Search
frontend
COMMON
CORE
HTML
CSS
CSS Frameworks
Programming Language
DB
SQL
DB Fundamentals
Data Modeling
Basic Programming
CODEVERSIONING
Basics - Version control fundamentals
Basics - Core Git commands (init, add, commit, push, pull)
Basics - Understanding Git file states (staged, modified, untracked)
Basics - .gitignore usage
UNITTESTING
Understanding Unit Testing Fundamentals - What is Unit Testing?
Understanding Unit Testing Fundamentals - Test-Driven Development (TDD)
Understanding Unit Testing Fundamentals - Anatomy of a Unit Test
Writing Effective Unit Tests - Test Case Design

Select Skill Group: Common

Select Developer Role: Frontend

5. Load Skill’s Matrix:

Navigate through Skill Matrix >> Followed by Designation (Ex: Junior Developer) and Role (Ex: Backend)



Skill Matrix Junior Developer - Backend

Select Tags: common

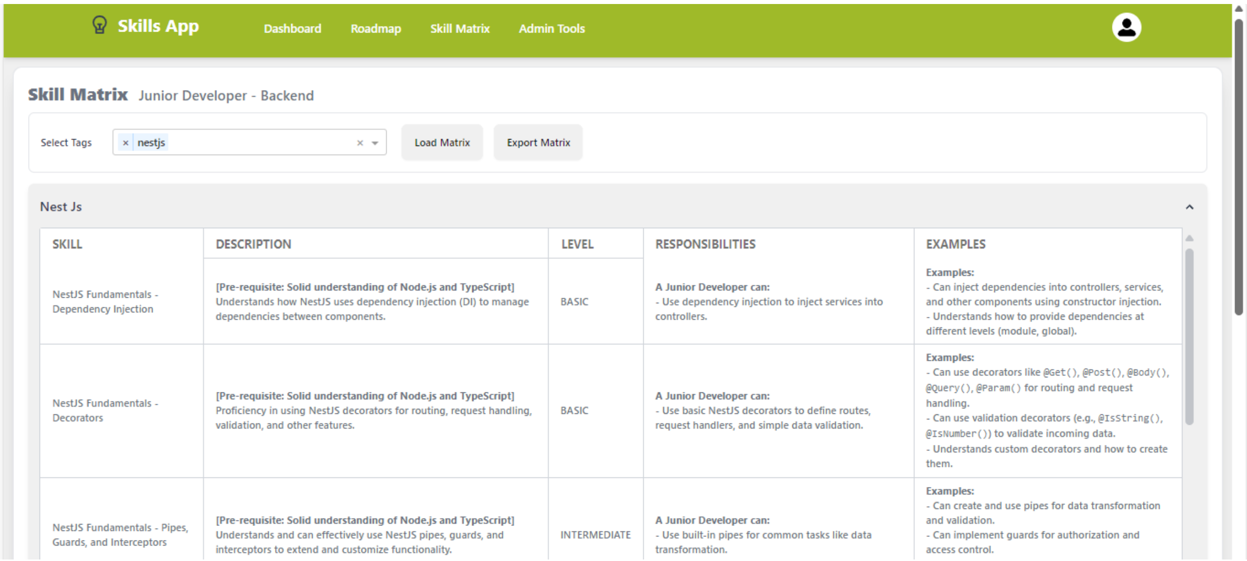
Skill's app allows the admin to load matrix of skills based on the tags passed as an argument . Enter tags of your desired skills.

Use the following tags for specific roles:

For Backend: keywords like : nestjs, nodejs, api, java, etc which are linked to backend work
For Frontend : keywords like: angular,vue.js,react.js etc which are linked to frontend work only

NOTE:- If you use any tags which are not corresponding to the role it will **NOT** load skills matrix for that particular tag.

Click on load matrix button to generate the skills matrix.



Skill Matrix Junior Developer - Backend

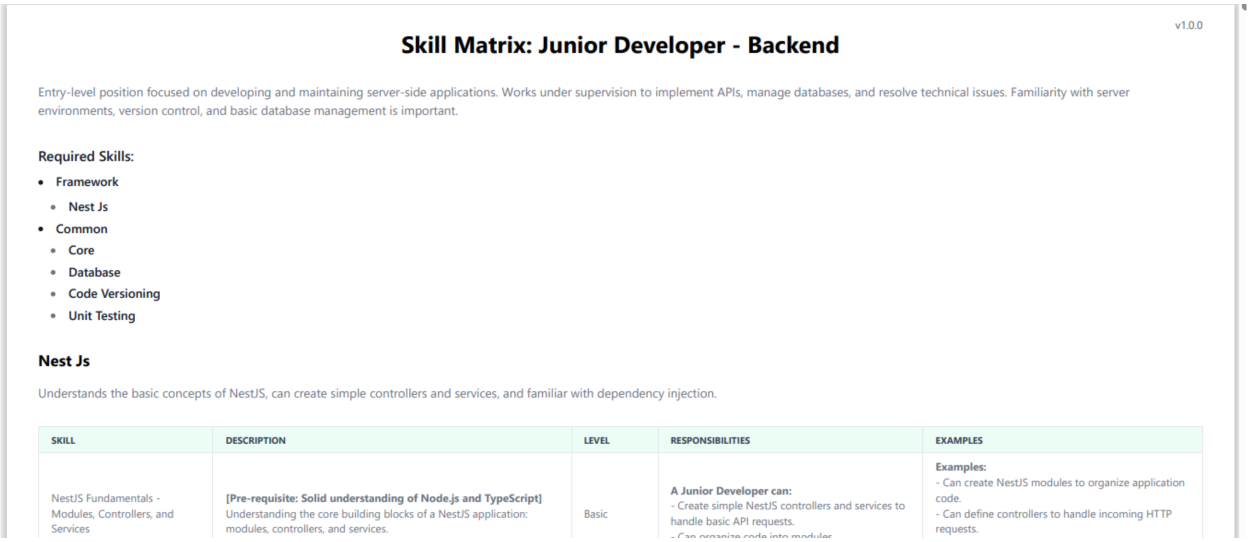
Select Tags: nestjs

Nest Js

SKILL	DESCRIPTION	LEVEL	RESPONSIBILITIES	EXAMPLES
NestJS Fundamentals - Dependency Injection	**[Pre-requisite: Solid understanding of Node.js and TypeScript]** Understands how NestJS uses dependency injection (DI) to manage dependencies between components.	BASIC	**A Junior Developer can:** - Use dependency injection to inject services into controllers.	**Examples:** - Can inject dependencies into controllers, services, and other components using constructor injection. - Understands how to provide dependencies at different levels (module, global).
NestJS Fundamentals - Decorators	**[Pre-requisite: Solid understanding of Node.js and TypeScript]** Proficiency in using NestJS decorators for routing, request handling, validation, and other features.	BASIC	**A Junior Developer can:** - Use basic NestJS decorators to define routes, request handlers, and simple data validation.	**Examples:** - Can use decorators like `@Get()`, `@Post()`, `@Body()`, `@Query()`, `@Param()` for routing and request handling. - Can use validation decorators (e.g., `@IsString()`, `@IsNumber()`) to validate incoming data. - Understands custom decorators and how to create them.
NestJS Fundamentals - Pipes, Guards, and Interceptors	**[Pre-requisite: Solid understanding of Node.js and TypeScript]** Understands and can effectively use NestJS pipes, guards, and interceptors to extend and customize functionality.	INTERMEDIATE	**A Junior Developer can:** - Use built-in pipes for common tasks like data transformation.	**Examples:** - Can create and use pipes for data transformation and validation. - Can implement guards for authorization and access control.

6. Export Matrix:

On clicking export matrix button, it allows admins to generate a pdf which will have all the information related to skills based on the tags passed



v1.0.0

Skill Matrix: Junior Developer - Backend

Entry-level position focused on developing and maintaining server-side applications. Works under supervision to implement APIs, manage databases, and resolve technical issues. Familiarity with server environments, version control, and basic database management is important.

Required Skills:

- **Framework**
  - **Nest Js**
- **Common**
  - **Core**
  - **Database**
  - **Code Versioning**
  - **Unit Testing**

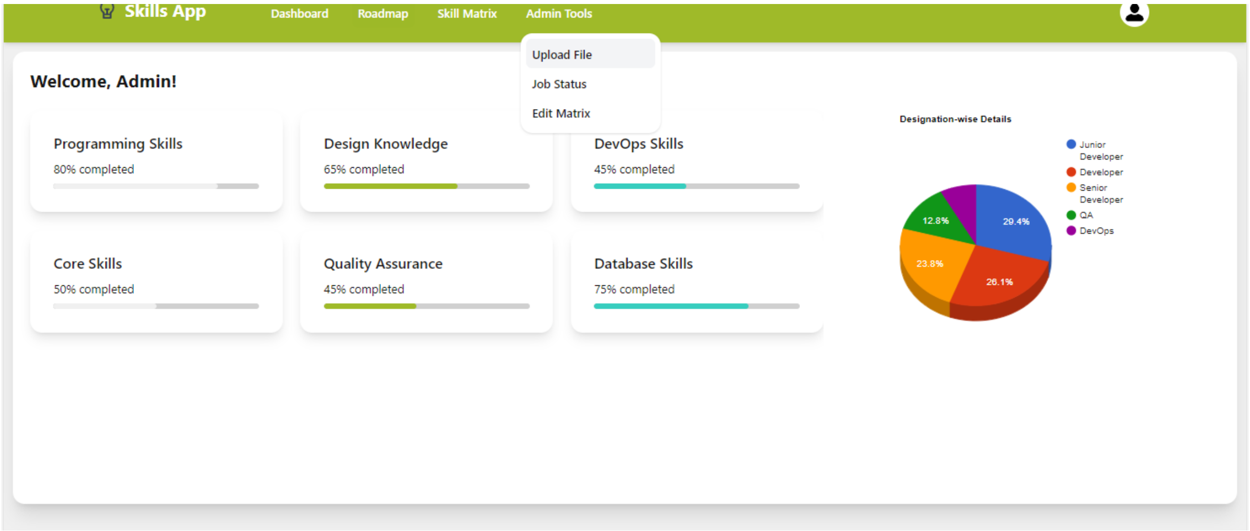
Nest Js

Understands the basic concepts of NestJS, can create simple controllers and services, and familiar with dependency injection.

SKILL	DESCRIPTION	LEVEL	RESPONSIBILITIES	EXAMPLES
NestJS Fundamentals - Modules, Controllers, and Services	**[Pre-requisite: Solid understanding of Node.js and TypeScript]** Understanding the core building blocks of a NestJS application: modules, controllers, and services.	Basic	**A Junior Developer can:** - Create simple NestJS controllers and services to handle basic API requests. - Can organize code into modules	**Examples:** - Can create NestJS modules to organize application code. - Can define controllers to handle incoming HTTP requests.

7. Edit Matrix:

Navigate through Admin tools >> Edit Matrix



Welcome, Admin!

Programming Skills
80% completed

Design Knowledge
65% completed

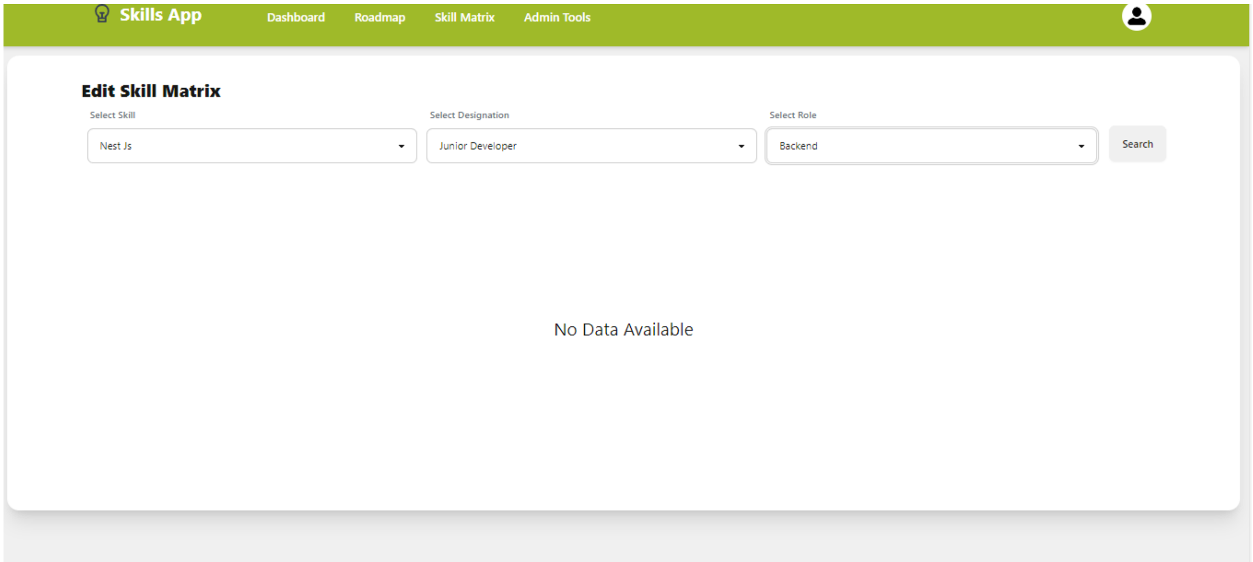
DevOps Skills
45% completed

Core Skills
50% completed

Quality Assurance
45% completed

Database Skills
75% completed

Click on the dropdown and select the desired skill,designation and role for the skills matrix



Edit Skill Matrix

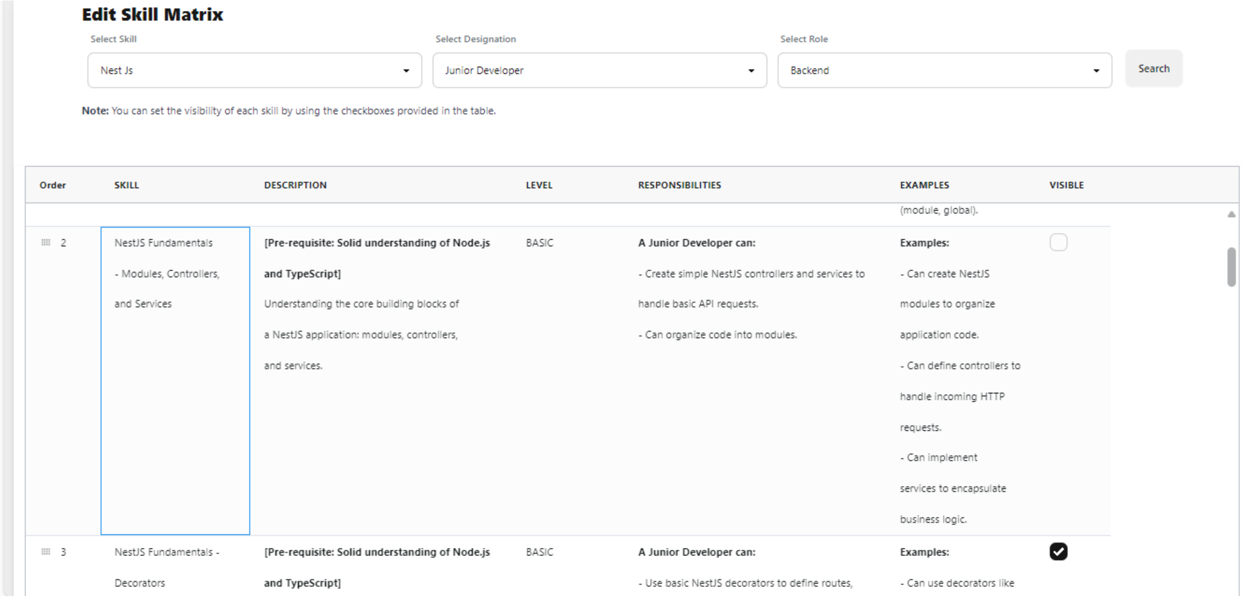
Select Skill: Nest Js

Select Designation: Junior Developer

Select Role: Backend

No Data Available

Skill's app allows the admin to change visibility and order number of the skills that would appear in the pdf when exported.



Edit Skill Matrix

Select Skill: Nest Js

Select Designation: Junior Developer

Select Role: Backend

Note: You can set the visibility of each skill by using the checkboxes provided in the table.

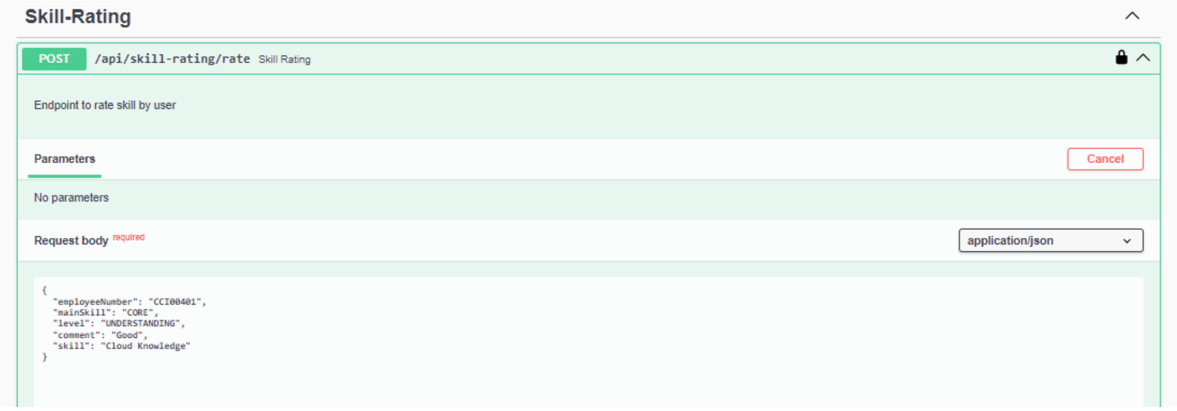
Order	SKILL	DESCRIPTION	LEVEL	RESPONSIBILITIES	EXAMPLES	VISIBLE
					(module, global).	
2	NestJS Fundamentals - Modules, Controllers, and Services	**[Pre-requisite: Solid understanding of Node.js and TypeScript]** Understanding the core building blocks of a NestJS application: modules, controllers, and services.	BASIC	**A Junior Developer can:** - Create simple NestJS controllers and services to handle basic API requests. - Can organize code into modules.	**Examples:** - Can create NestJS modules to organize application code. - Can define controllers to handle incoming HTTP requests. - Can implement services to encapsulate business logic.	☐
3	NestJS Fundamentals - Decorators	**[Pre-requisite: Solid understanding of Node.js and TypeScript]**	BASIC	**A Junior Developer can:** - Use basic NestJS decorators to define routes,	**Examples:** - Can use decorators like	☑

This allows admins to have matrices of different order and hiding some rows in the table as per required

The admin can go back to Load Matrix option to check if their changes have been reflected

8. Skill Rating:

The logged in user can rate themselves for a skill which is mentioned



Skill-Rating

/api/skill-rating/rate Skill Rating

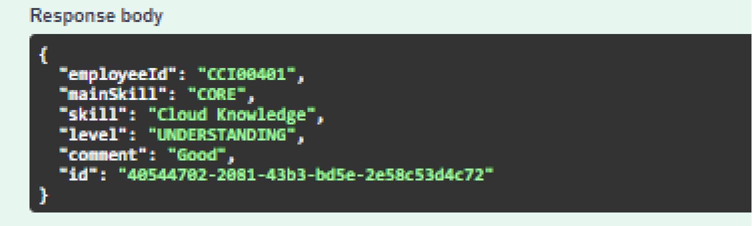
Endpoint to rate skill by user

Parameters

No parameters

Request body required

```
{
  "employeeNumber": "CCI00401",
  "mainSkill": "CORE",
  "level": "UNDERSTANDING",
  "comment": "Good",
  "skill": "Cloud Knowledge"
}
```



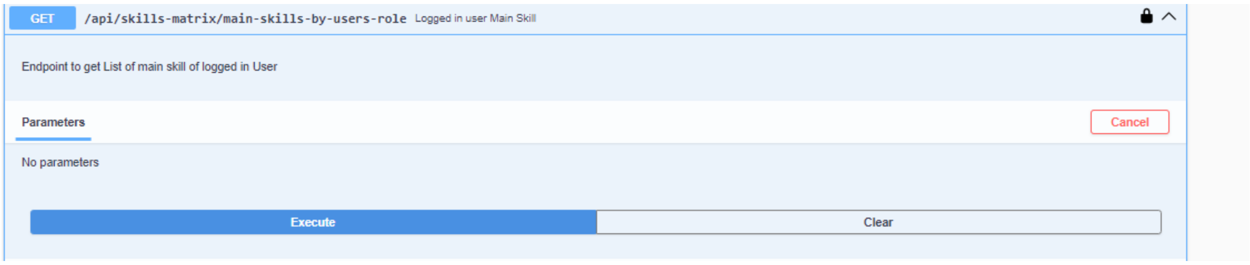
Response body

```
{
  "employeeId": "CCI00401",
  "mainSkill": "CORE",
  "skill": "Cloud Knowledge",
  "level": "UNDERSTANDING",
  "comment": "Good",
  "id": "40544702-2081-43b3-bd5e-2e58c53d4c72"
}
```

9. Skill Matrix for LoggedIn User:

a. Get Main Skills

A logged in user can get all the available main skills based on their role in the employee table (ex Backend,Frontend, QA)

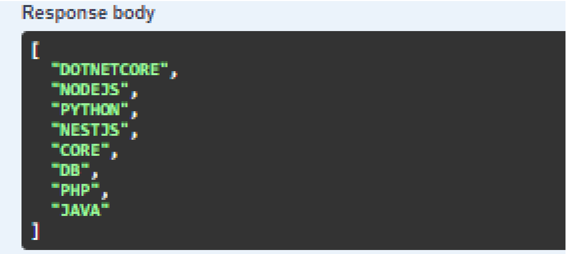


/api/skills-matrix/main-skills-by-users-role Logged in user Main Skill

Endpoint to get List of main skill of logged in User

Parameters

No parameters

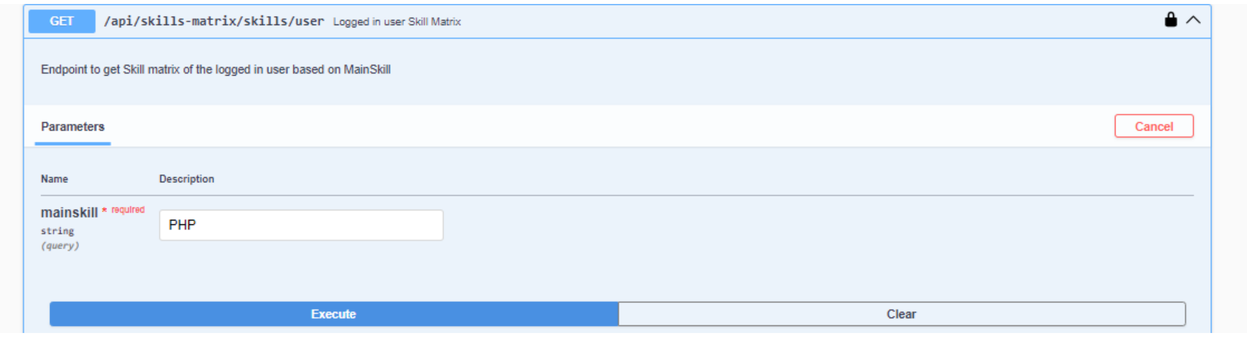


Response body

```
[
  "DOTNETCORE",
  "NODEJS",
  "PYTHON",
  "NESTJS",
  "CORE",
  "DB",
  "PHP",
  "JAVA"
]
```

b. Generate Skill’s Matrix

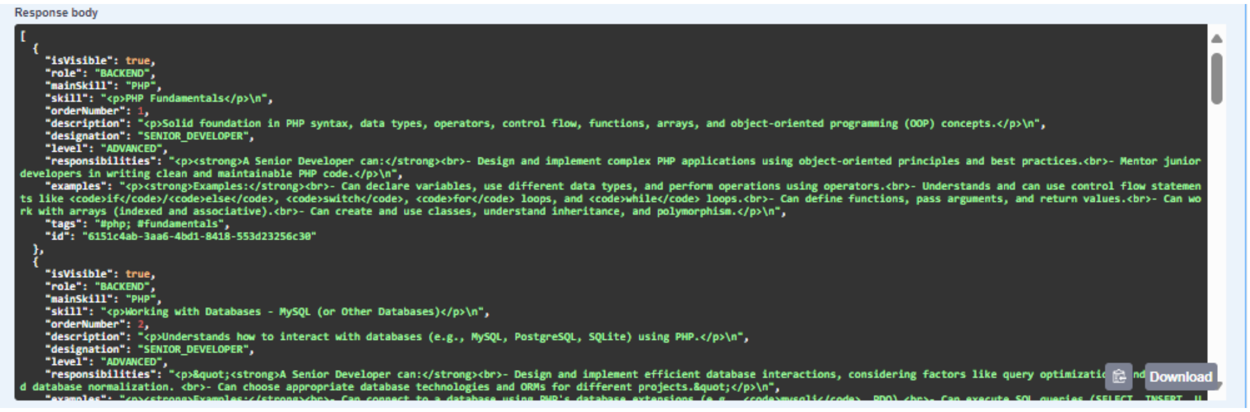
A logged in user can generate their skill matrix based on their main skills



/api/skills-matrix/skills/user Logged in user Skill Matrix

Endpoint to get Skill matrix of the logged in user based on MainSkill

Name	Description
mainskill * required string (query)	PHP

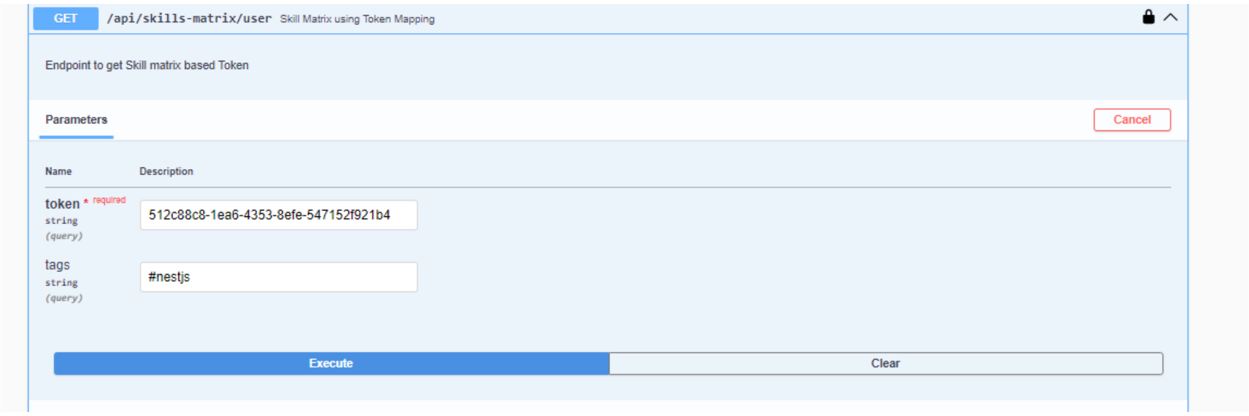


Response body

```
[
  {
    "isVisible": true,
    "role": "BACKEND",
    "mainSkill": "PHP",
    "skill": "<p>PHP Fundamentals</p>\n",
    "orderNumber": 1,
    "description": "<p>Solid foundation in PHP syntax, data types, operators, control flow, functions, arrays, and object-oriented programming (OOP) concepts.</p>\n",
    "designation": "SENIOR_DEVELOPER",
    "level": "ADVANCED",
    "responsibilities": "<p><strong>A Senior Developer can:</strong><br>- Design and implement complex PHP applications using object-oriented principles and best practices.<br>- Mentor junior
developers in writing clean and maintainable PHP code.</p>\n",
    "examples": "<p><strong>Examples:</strong><br>- Can declare variables, use different data types, and perform operations using operators.<br>- Understands and can use control flow statemen
ts like <code>if</code>/<code>else</code>, <code>switch</code>, <code>for</code> loops, and <code>while</code> loops.<br>- Can define functions, pass arguments, and return values.<br>- Can wo
rk with arrays (indexed and associative).<br>- Can create and use classes, understand inheritance, and polymorphism.</p>\n",
    "tags": "#php; #fundamentals",
    "id": "6151c4ab-3aa6-4bd1-8418-553d23256c30"
  },
  {
    "isVisible": true,
    "role": "BACKEND",
    "mainSkill": "PHP",
    "skill": "<p>Working with Databases - MySQL (or Other Databases)</p>\n",
    "orderNumber": 2,
    "description": "<p>Understands how to interact with databases (e.g., MySQL, PostgreSQL, SQLite) using PHP.</p>\n",
    "designation": "SENIOR_DEVELOPER",
    "level": "ADVANCED",
    "responsibilities": "<p>&quot;<strong>A Senior Developer can:</strong><br>- Design and implement efficient database interactions, considering factors like query optimizatic nd
d database normalization. <br>- Can choose appropriate database technologies and ORMs for different projects.&quot;</p>\n",
    "examples": "<p><strong>Examples:</strong><br>- Can connect to a database using PHP's database extensions (e.g., <code>mysqli</code>, PDO).<br>- Can execute SQL queries (SELECT, INSERT, U
```

10. Skill matrix based on Token:

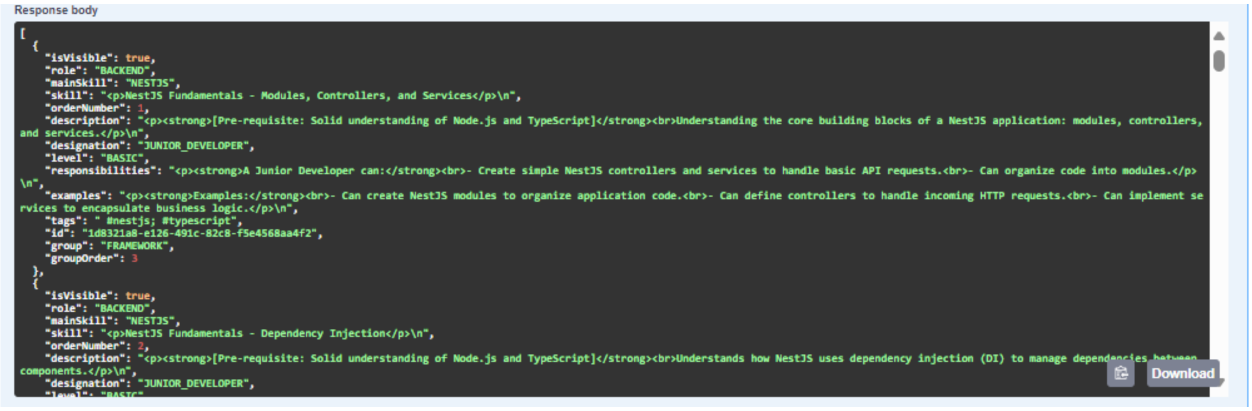
A link with token which is based on some designation and role can be shared with anonymous person who will be directed to a page where after adding the tags required skill matrix will be generated



/api/skills-matrix/user Skill Matrix using Token Mapping

Endpoint to get Skill matrix based Token

Name	Description
token * required string *(query)*	512c88c8-1ea6-4353-8efe-547152f921b4
tags string *(query)*	#nestjs


Response body

```
[
  {
    "isVisible": true,
    "role": "BACKEND",
    "mainSkill": "NESTJS",
    "skill": "<p>NestJS Fundamentals - Modules, Controllers, and Services</p>\n",
    "orderNumber": 1,
    "description": "<p><strong>[Pre-requisite: Solid understanding of Node.js and TypeScript]</strong><br>Understanding the core building blocks of a NestJS application: modules, controllers,
and services.</p>\n",
    "designation": "JUNIOR_DEVELOPER",
    "level": "BASIC",
    "responsibilities": "<p><strong>A Junior Developer can:</strong><br>- Create simple NestJS controllers and services to handle basic API requests.<br>- Can organize code into modules.</p>
\n",
    "examples": "<p><strong>Examples:</strong><br>- Can create NestJS modules to organize application code.<br>- Can define controllers to handle incoming HTTP requests.<br>- Can implement se
rvices to encapsulate business logic.</p>\n",
    "tags": " #nestjs; #typescript",
    "id": "1d8321a8-e126-491c-82c8-f5e4568aa4f2",
    "group": "FRAMEWORK",
    "groupOrder": 3
  },
  {
    "isVisible": true,
    "role": "BACKEND",
    "mainSkill": "NESTJS",
    "skill": "<p>NestJS Fundamentals - Dependency Injection</p>\n",
    "orderNumber": 2,
    "description": "<p><strong>[Pre-requisite: Solid understanding of Node.js and TypeScript]</strong><br>Understands how NestJS uses dependency injection (DI) to manage dependencies between
components.</p>\n",
    "designation": "JUNIOR_DEVELOPER",
    "level": "BASIC"
```