Project title: Media Streaming using Cloud

Problem Statement : Create a virtual cinema platform using IBM Cloud Video Streaming. Upload and stream your favourite movies and videos on-demand. Share the joy of movie nights with friends and family, no matter where they are located. Elevate the movie-watching experience with seamless streaming and high- quality video playback for a truly immersive cinematic experience!.

Phase 4: development part 2

In this part we have to begin building our project. We have to start building the virtual cinema platform using ibm cloud video streaming.

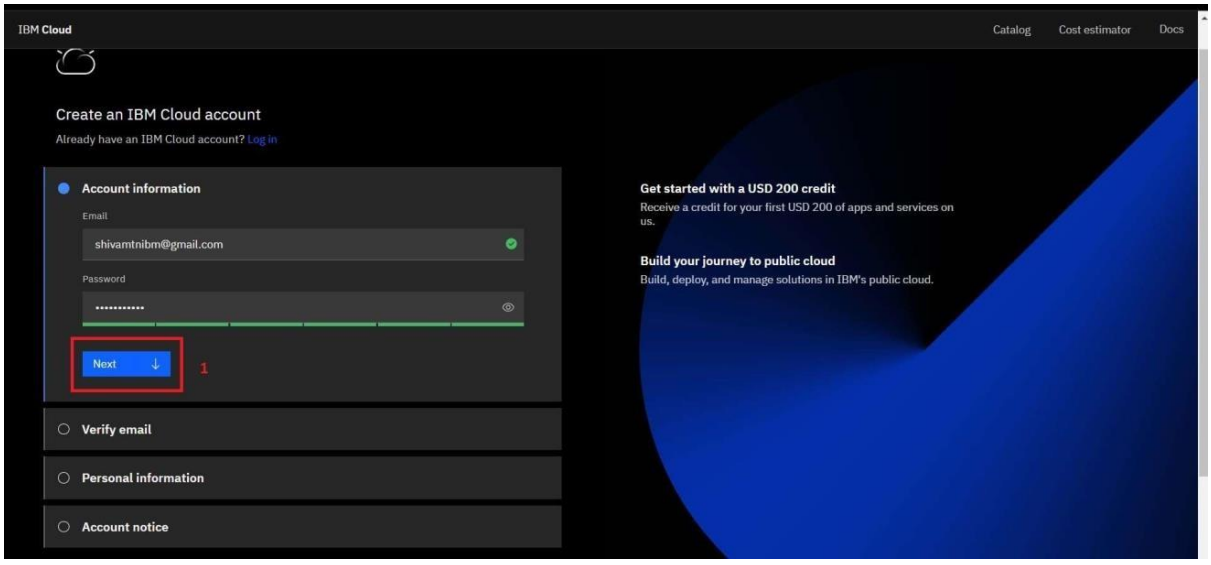
Steps:

1. Create an ibm account.
2. Platform's features and design.
3. User registration and authentication mechanisms.

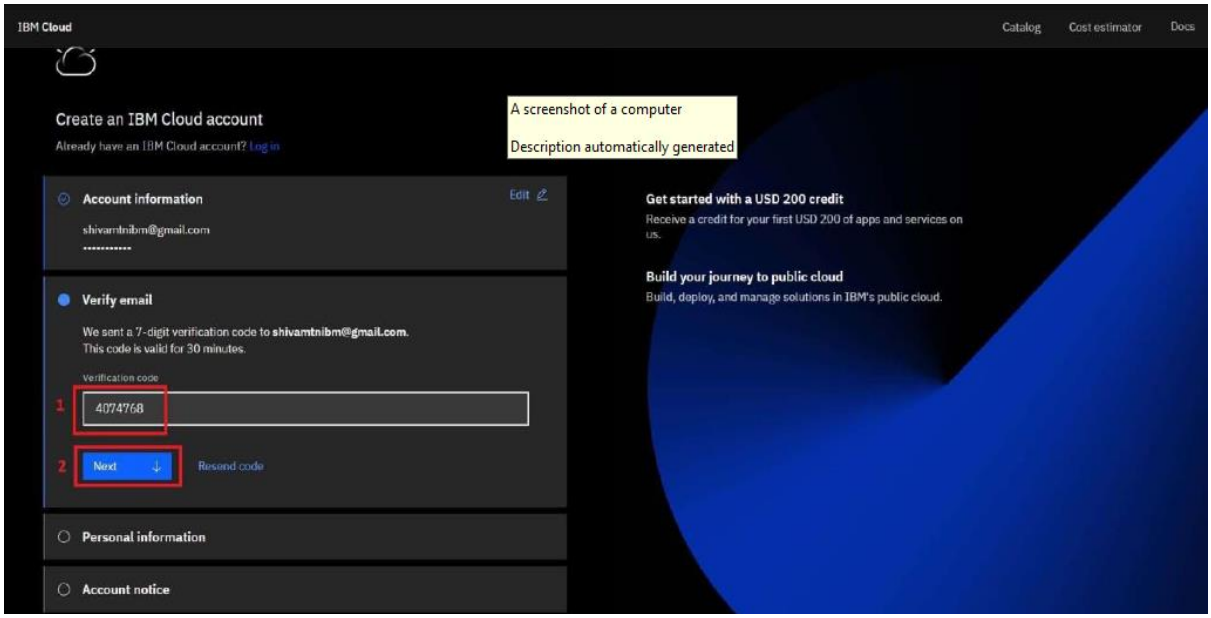
Implementation:

1.creating an ibm account

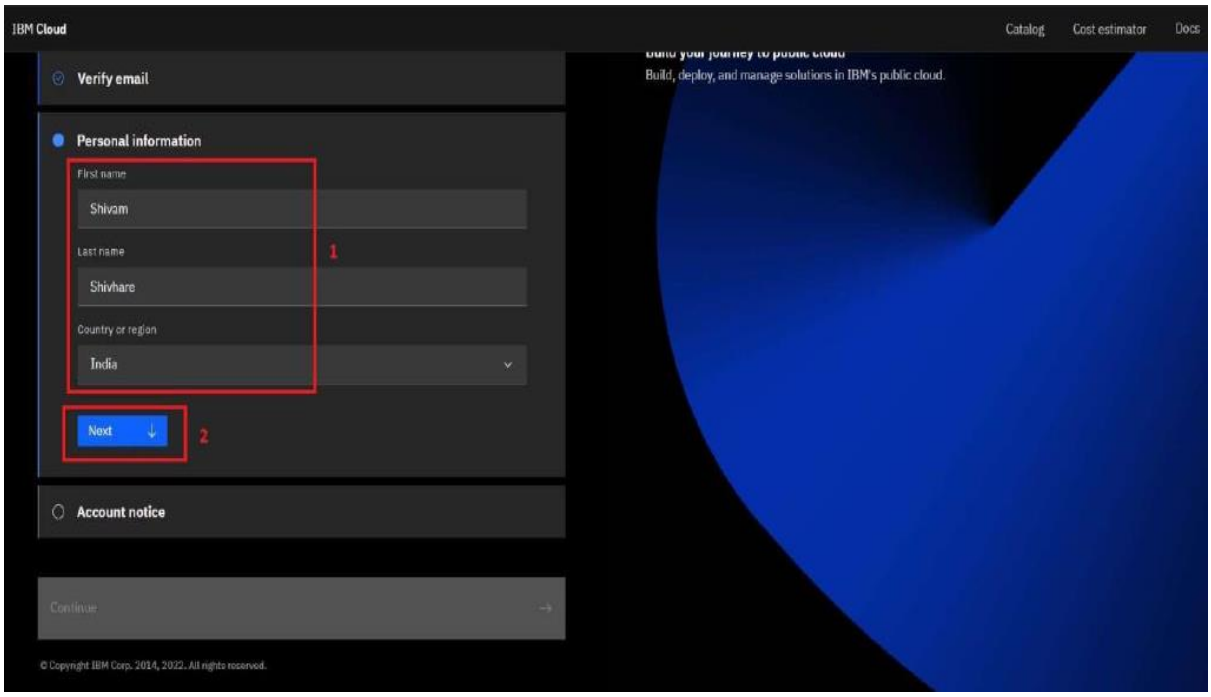
using the ibm feature code provide in the skill up online create an ibm account. First step for the account creation is the account information, provide this space with the email address and a specific password for the ibm account.



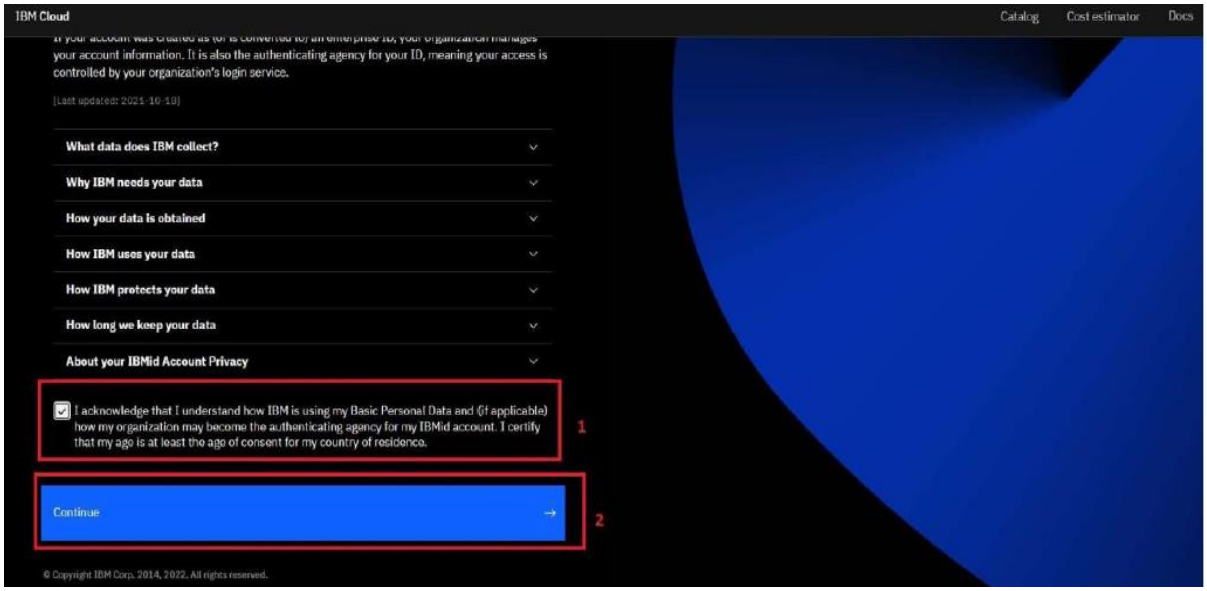
after providing the email address and the password a verifictaion code will be sent to the gmail account provide the code obtained form the mail in the box given and verify your account.



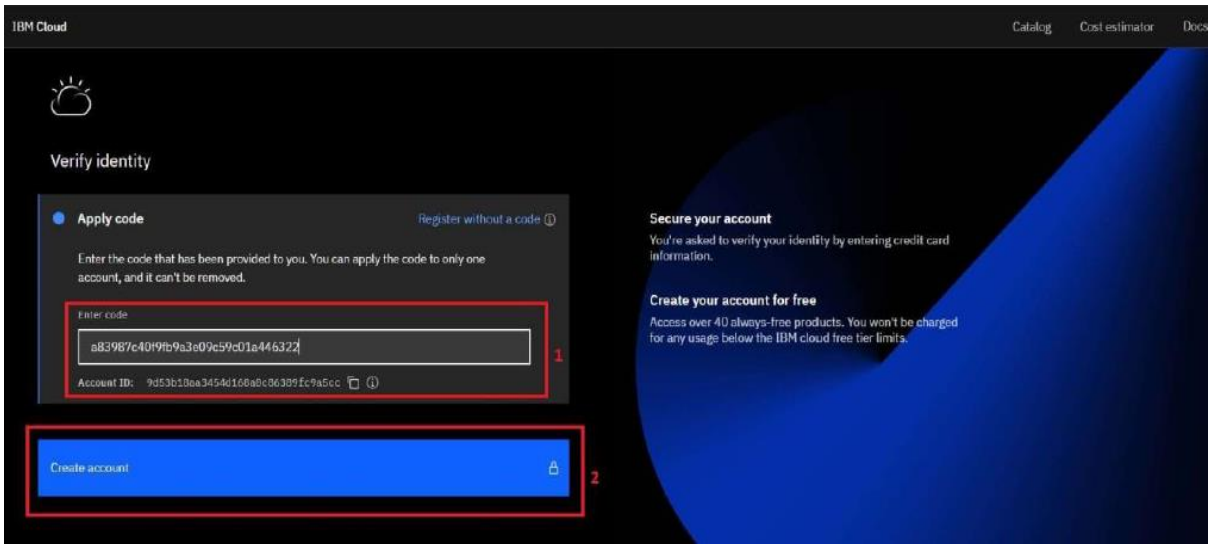
after account verification provide your personal information such as first name,last name and country.



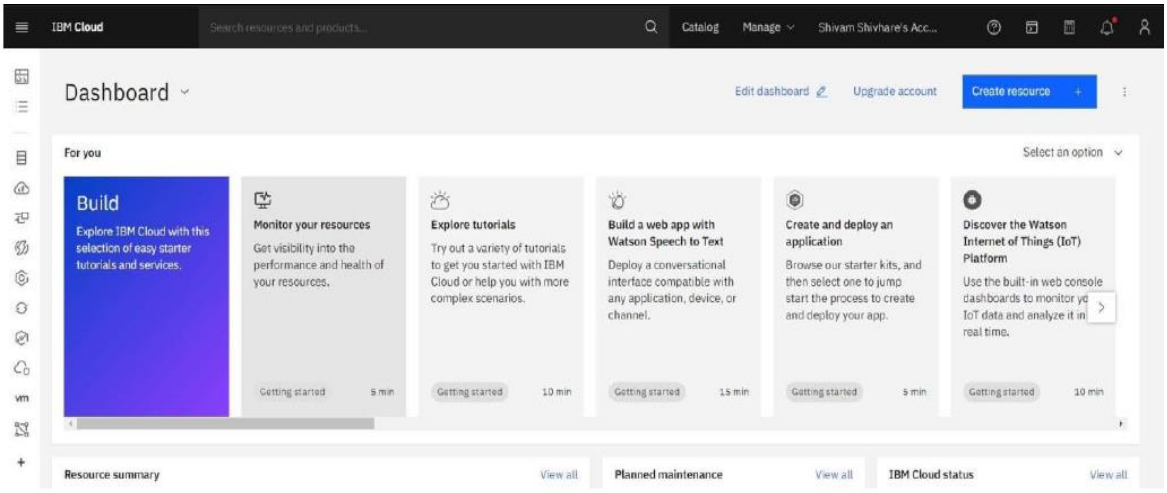
accept the terms and conditions provided by the ibm cloud after reading them carefully. Click the continue button to go to the next step.



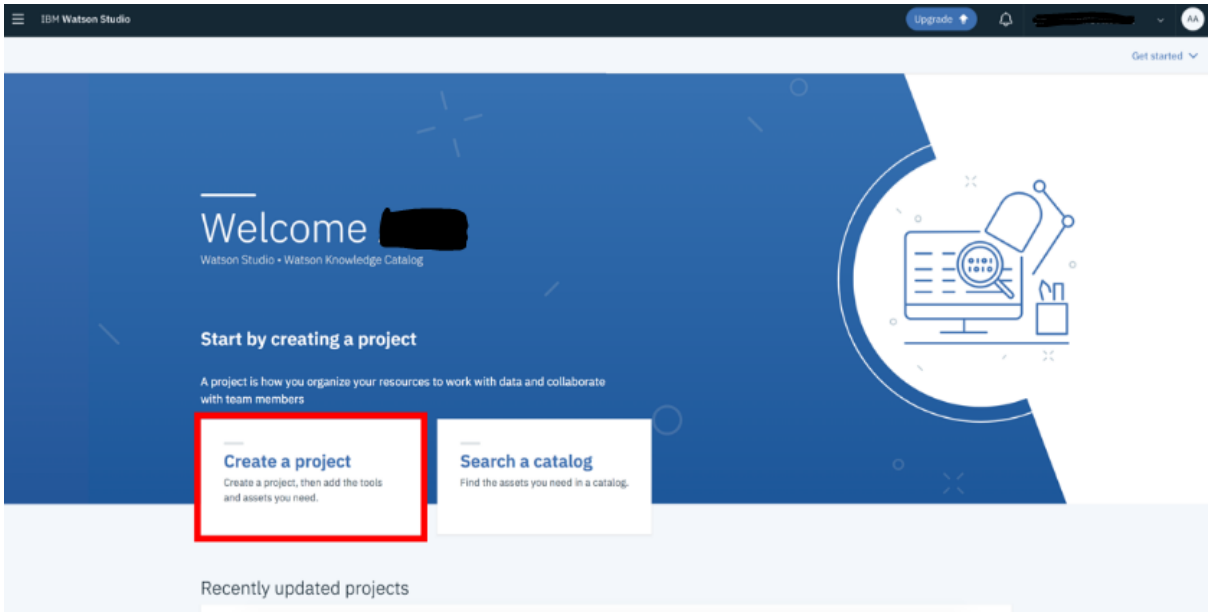
provide the ‘enter code’ section with the trial code obtained from the skill up account



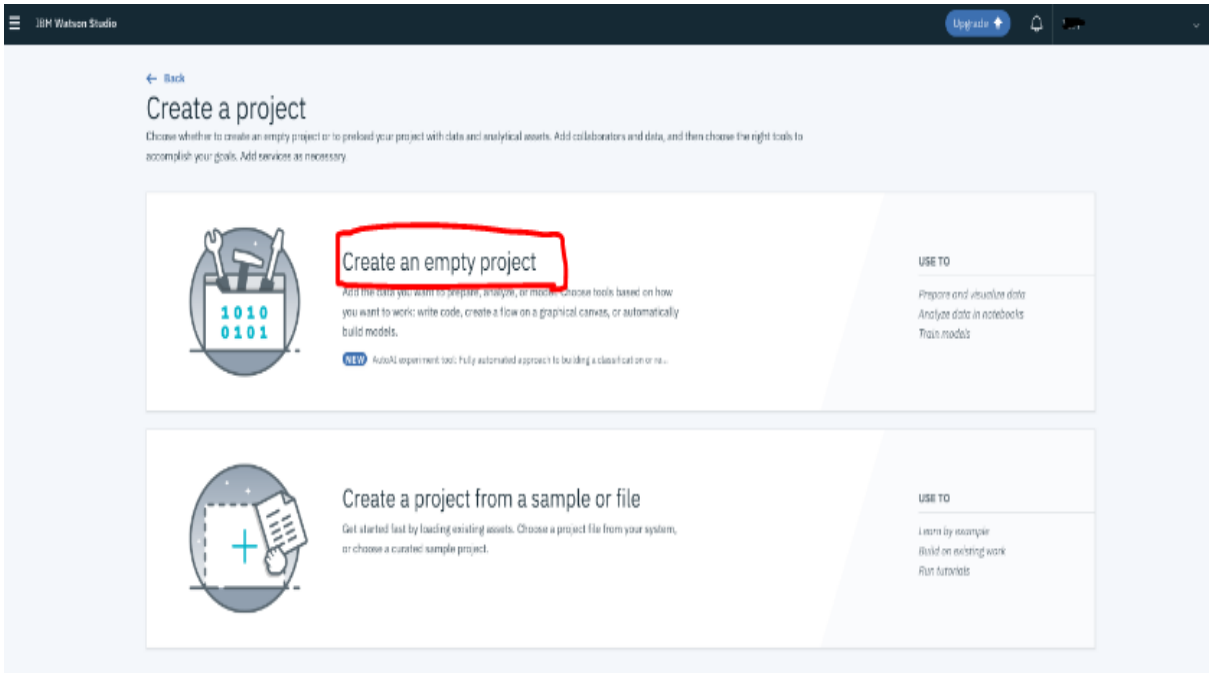
after the account creation process you will be led to the dashboard.



Once you land on the ibm watson studio main page, start by creating a project.

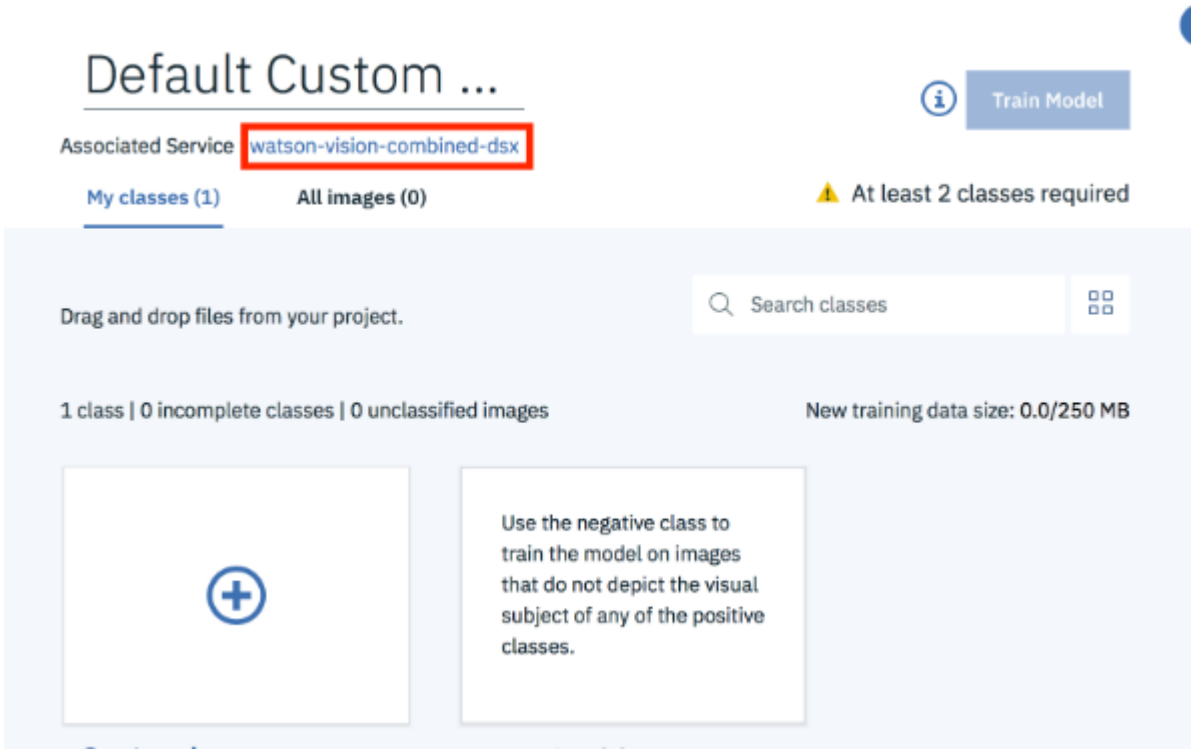


Choose create empty project.



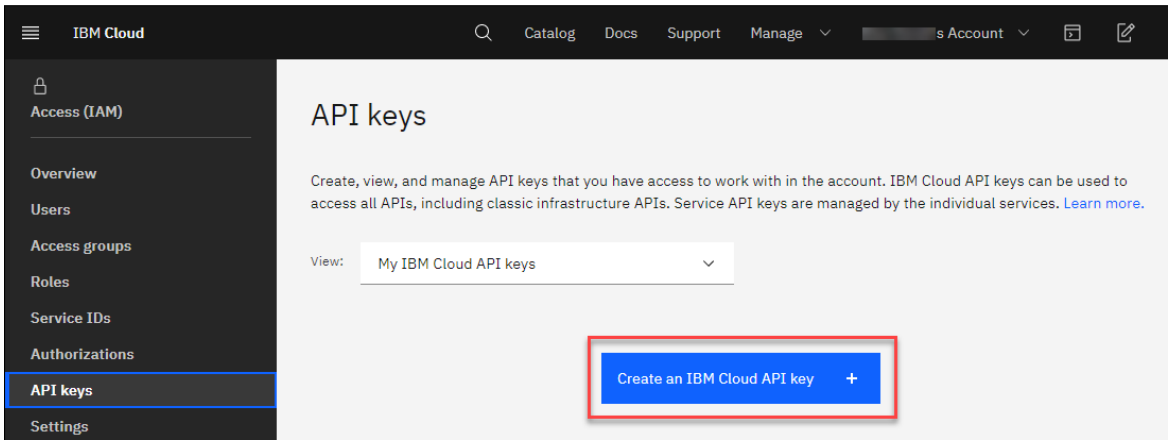
Go to add to project and choose image classification.

Select built-in models for watson visual recognition.

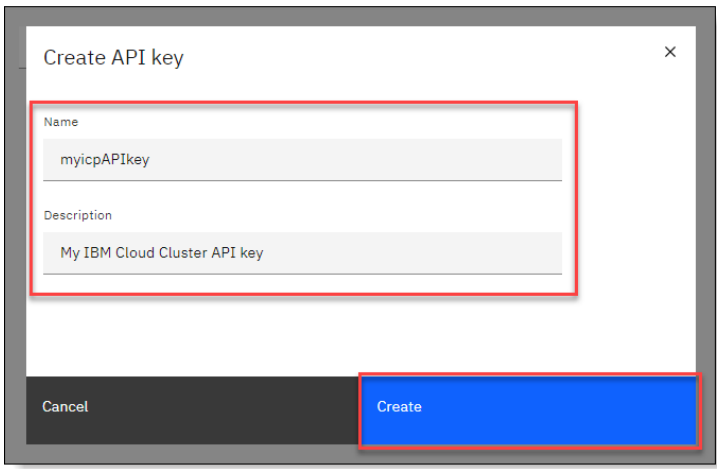


Obtaining api keys

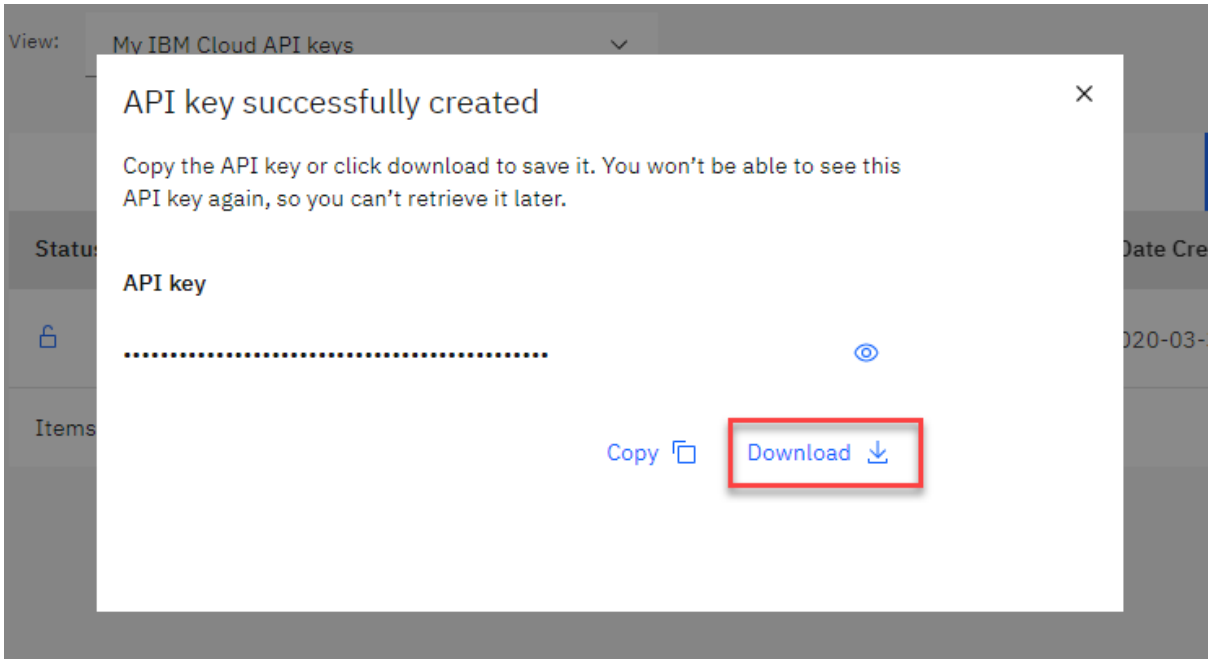
after setting up the watson visual recognition in ibm cloud go to the api keys section. In the api section click on the create an ibm cloud api key button



after clicking on the create button give fill the boxes provided with the name and the description of the image that you are going to classify.



After providing all the necessary details click on the create button. Click download



thus the development part 1 for the image recognition using ibm cloud visual recognition is done and documented successfully.

Creating a virtual cinema platform using ibm cloud video streaming involves setting up and configuring the necessary components to host and stream movies or other video content to your audience. Below, i'll outline the steps to create such a platform using ibm cloud video streaming:

1. Create an ibm cloud video streaming account:

- log in to your ibm cloud account.

- navigate to the ibm cloud video streaming service and create a new instance.

2. Set up your video content:

- upload the movies or videos you want to stream to the ibm cloud video streaming platform.

- ibm cloud video streaming provides features to organize, categorize, and manage your video content.

3. Create and configure a channel:

- create channels for streaming your movies or content. Channels act as virtual screening rooms.

- configure settings for each channel, including privacy options, playback settings, and access controls.

4. Integrate video player:

- embed the ibm cloud video streaming player into your website or application to allow users to watch the content.

5. Monetization (optional):

- if you want to charge users for access to your virtual cinema platform, consider integrating payment gateways or subscription models. Ibm cloud video streaming may offer options for monetization.

6. Security and access control:

- implement security measures such as token-based authentication, geo-restrictions, and user access control to protect your content.

7. Custom branding:

- customize the video player and platform with your branding, including logos, colors, and themes.

8. User engagement features:

- implement interactive features such as live chat, comments, or polls to enhance user engagement.

9. Analytics and reporting:

- use analytics tools provided by ibm cloud video streaming to track user engagement, view counts, and other important metrics.

10. Testing and quality assurance:

- test the platform extensively to ensure the streaming quality and user experience meet your standards.

11. Launch and promotion:

- once you're satisfied with the platform's performance, launch your virtual cinema platform and promote it to your target audience.

12. Support and maintenance:

- provide customer support and regular maintenance to ensure the platform runs smoothly.

Please note that ibm cloud video streaming capabilities and features may change over time, so it's essential to refer to the most recent documentation and support resources provided by ibm for up-to-date guidance on setting up your virtual cinema platform.

Designing an intuitive user interface for a virtual cinema platform using ibm cloud video streaming requires careful consideration of user experience, content organization, and ease of navigation. Here's a step-by-step guide on how to create such a ui:

Header: place the platform's logo, navigation menu, and user profile options here.

Featured content: showcase top movies or events in a visually appealing slider or grid format.

Search bar: include a prominent search bar for users to find movies, events, or genres.

Categories: display popular movie genres (e.g., action, drama, comedy) as clickable tiles.

2. Movie/event page:

title and description: show the movie/event title and a brief description.

poster/thumbnail: display an eye-catching image.

play button: large and conspicuous play button to start streaming.

ratings and reviews: add ratings, user reviews, and comments.

share and save: buttons for sharing on social media and saving to a watchlist.

4. Streaming page:

Video player: central video player with playback controls (play, pause, volume,

Fullscreen).

Subtitles: option to enable/disable subtitles.

Quality settings:a menu for adjusting streaming quality.

related content: suggest similar movies/events the user might like.

user profile access: access user settings, watchlist, and payment options.

4. User profile:

profile picture: allow users to upload a profile picture.

watchlist: show movies/events users have saved to watch later.

purchase history: display past transactions and purchases.

payment settings: options for updating payment methods.

5. Search and browse:

search results: show a grid of search results with movie/event thumbnails and titles.

filter and sort: provide options to filter by genre, release date, rating, and sorting criteria.

pagination: if there are many results, offer pagination for easier navigation.

6. Payment and subscriptions:

pricing plans: clearly present subscription options (monthly, yearly) and pay-per-view prices.

payment gateway: provide a secure and straightforward payment process.

coupons/promotions: allow users to redeem coupons or enter promo codes.

7. Settings:

Account settings: options to change passwords, email, or personal information.

Notification preferences: allow users to manage email and in-app notifications.

Language and accessibility: offer language settings and accessibility options like closed captions.

8. Support and help:

Faq and help center: link to a comprehensive faq section or knowledge base.

Contact support: provide a contact form or chat for customer support.

Feedback: encourage users to provide feedback for improvements.

9. Mobile responsiveness:

- ensure the platform is mobile-friendly with responsive design.
- consider a mobile app for a better mobile viewing experience.

10. Consistent design elements:

- use a consistent color scheme, fonts, and layout throughout the platform.
- make buttons and interactive elements intuitive and easily clickable.

11. User onboarding:

- offer a guided tour for first-time users to explain how the platform works.

12. Security:

- prioritize security for payment transactions and user data.

Remember to conduct user testing and gather feedback during the design and development process to ensure that the user interface is genuinely intuitive and meets the needs of your target audience. Additionally, consider a/b testing to fine-tune the platform based on user preferences and behavior.

Setting up user registration and authentication mechanisms is crucial for ensuring secure access to a platform. Below is a general guide to help you get started. Keep in mind that the specific implementation details may vary depending on your platform, programming language, and framework.

1. Choose a technology stack:

determine the programming language and framework you will use for your platform. Common choices include node.js with express, ruby on rails, django, or asp.net core.

2. Database setup:

set up a database to store user information securely. You can use relational databases like mysql, postgresql, or nosql databases like mongodb.

3.user registration:

create a user registration system to allow users to sign up for your platform. Here are the basic steps:

- create a registration form that collects necessary user information, such as username, email, and password.

- validate and sanitize user inputs to prevent common security issues like sql injection and cross-site scripting (xss).

- hash and salt the user's password before storing it in the database to enhance security. Libraries like bcrypt can help with this.

- store the user's information in the database.

4. User authentication:

implement authentication to verify the identity of users. Here are the steps:

- create a login system with a form that accepts the user's credentials (username/email and password).

- verify the user's credentials by comparing the provided password with the stored and hashed password in the database.

- generate a session token or json web token (jwt) upon successful authentication and store it in a secure http cookie.

- use the session token or jwt for subsequent requests to authenticate users.

5. Password recovery:

implement a password recovery mechanism, such as a "forgot password" feature. This typically involves sending a password reset link to the user's email.

6. Security best practices:

ensure that your registration and authentication systems follow security best practices:

- implement rate limiting and account lockout to protect against brute-force attacks.

- use https to encrypt data transmission, preventing eavesdropping.

- apply input validation and output encoding to prevent common web vulnerabilities.

- keep your software and libraries up to date to patch security vulnerabilities.

7. User roles and permissions:

implement a role-based access control system to manage user permissions. Define roles (e.g., admin, user) and restrict access to certain functionalities based on the user's role.

8. Logging and monitoring:

set up logging and monitoring to track and respond to suspicious activities, such as failed login attempts or unauthorized access.

9. Two-factor authentication (2fa):

consider implementing 2fa for an extra layer of security, requiring users to enter a one-time code sent to their mobile device or email.

10. Third-party authentication:

allow users to log in using third-party providers like google, facebook, or github. This can improve user experience and security.

11. Testing:

thoroughly test your registration and authentication systems, including security testing like penetration testing and code review.

12. Compliance:

ensure your system complies with relevant regulations, such as gdpr or hipaa, if applicable.

13. Documentation:

document your registration and authentication processes for developers and administrators. This includes documenting the security measures in place.

14. User education:

educate users about best practices for password security and online safety.

15. Continual improvement:

regularly update and improve your authentication and security mechanisms to adapt to evolving threats and best practices.remember that security is an ongoing process. Regularly monitor and update your systems to stay ahead of emerging threats.

Coding part :

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Filmlane - Best movie collections</title>
 <!-- - favicon -->
 <link rel="shortcut icon" href="./favicon.svg" type="image/svg+xml">
<!-- - custom css link-->
 <link rel="stylesheet" href="./assets/css/style.css">
<!-- - google font link -->
 <link rel="preconnect" href="https://fonts.googleapis.com">
 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
 <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap"
rel="stylesheet"> </head>
<body id="top">
<!--- #HEADER-->
<header class="header" data-header>
 <div class="container">
<div class="overlay" data-overlay></div>
```

```
<a href="./index.html" class="logo">
<img src="./assets/images/logo.svg" alt="Filmlane logo">
</a>
<div class="header-actions">
<button class="search-btn">
<ion-icon name="search-outline"></ion-icon>
</button>
<div class="lang-wrapper">
<label for="language">
<ion-icon name="globe-outline"></ion-icon>
</label>
<select name="language" id="language">
<option value="en">EN</option>
<option value="au">AU</option>
<option value="ar">AR</option>
<option value="tu">TU</option>
</select>
</div>
<button class="btn btn-primary">Sign in</button>
</div>
<button class="menu-open-btn" data-menu-open-btn>
<ion-icon name="reorder-two"></ion-icon>
</button>
<nav class="navbar" data-navbar>
<div class="navbar-top">
<a href="./index.html" class="logo">
<img src="./assets/images/logo.svg" alt="Filmlane” </a> <button class="menu-close
```

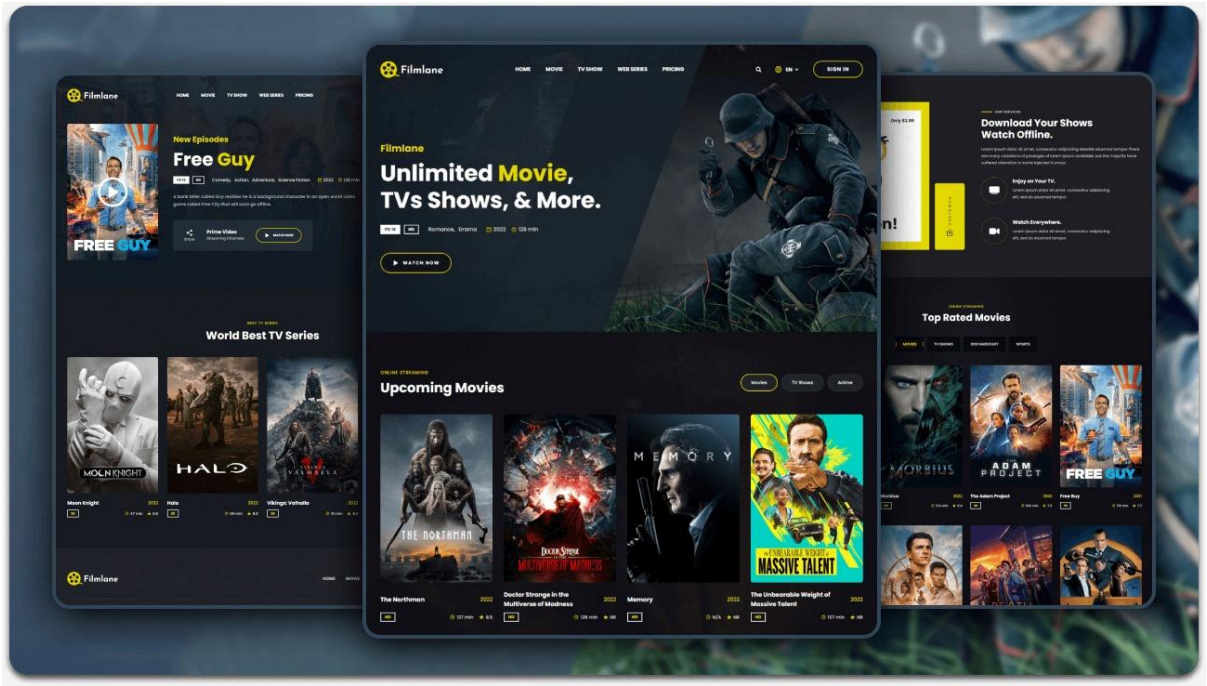
```
<ion-icon name="close-outline"></ion-icon>
 </button>
 </div>
 <ul class="navbar-list">
 <li>
 <a href="./index.html" class="navbar-link">Home</a>
 </li>
 <li>
 <a href="#" class="navbar-link">Movie</a>
 </li>
 <li>
 <a href="#" class="navbar-link">Tv Show</a>
 </li>
 <li>
 <a href="#" class="navbar-link">Web Series</a>
 </li>
 <li>
 <a href="#" class="navbar-link">Pricing</a>
 </li>
 </ul>
 <ul class="navbar-social-list">
 <li>
 <a href="#" class="navbar-social-link">
 <ion-icon name="logo-twitter"></ion-icon>
 </a>
 </li>
 <li>
```

```
<a href="#" class="navbar-social-link">
<ion-icon name="logo-facebook"></ion-icon>
<li>
<a href="#" class="navbar-social-link">
<ion-icon name="logo-pinterest"></ion-icon>
</a>
</li>
<li>
<a href="#" class="navbar-social-link">
<ion-icon name="logo-instagram"></ion-icon>
</a>
</li>
<li>
<a href="#" class="navbar-social-link">
<ion-icon name="logo-youtube"></ion-icon>
</a>
</li>
</ul>
</nav>
</div>
</header>
<main>
<article>
<!--
- #HERO
-->
<section class="hero">
```

```
<div class="container">
<div class="hero-content">
<p class="hero-subtitle">Filmlane</p>
<h1 class="h1 hero-title">
Unlimited <strong>Movie</strong>, TVs Shows, & More. </h1>
<div class="meta-wrapper">
<div class="badge-wrapper">
<div class="badge badge-fill">PG 18</div>
<div class="badge badge-outline">HD</div>
</div>
<div class="ganre-wrapper">
<a href="#">Romance,</a>
<a href="#">Drama</a>
</div>
<div class="date-time">
<div>
<ion-icon name="calendar-outline"></ion-icon>
<time datetime="2022">2022</time>
</div>
<div>
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT128M">128 min</time>
</div>
</div>
</div>
<button class="btn btn-primary">
<ion-icon name="play"></ion-icon>
```

```
<span>Watch now</span>

</button>

</div>

</div>

</section>

<!-- - #UPCOMING -->

<section class="upcoming">

<div class="container">

<div class="flex-wrapper">

<div class="title-wrapper">

[9:24 pm, 26/10/2023] Mani: </div>

<div class="rating">

<ion-icon name="star"></ion-icon>

<data>7.8</data>

</div>

</div>

</div>

</li>

<li>

<div class="movie-card">

<a href="./movie-details.html">

<figure class="card-banner">

<img src="./assets/images/movie-2.png" alt="Morbius movie poster">

</figure>

</a>

<div class="title-wrapper">

<a href="./movie-details.html">
```

```
<h3 class="card-title">Morbius</h3>

</a>

<time datetime="2022">2022</time>

</div>

<div class="card-meta">

<div class="badge badge-outline">HD</div>

<div class="duration">

<ion-icon name="time-outline"></ion-icon>

<time datetime="PT104M">104 min</time>

</div>

<div class="rating">

<ion-icon name="star"></ion-icon>

<data>5.9</data>

</div>

[9:24 pm, 26/10/2023] Mani: </div>

</div>

</li>

<li>

<div class="movie-card">

<a href="./movie-details.html">

<figure class="card-banner">

<img src="./assets/images/movie-3.png" alt="The Adam Project movie poster">

</figure>

</a>
```



Conclusion:

IBM Cloud offers a comprehensive suite of media streaming services that can help businesses of all sizes reach a global audience with high-quality video content. IBM Cloud Video Streaming is a cloud-based video streaming platform that offers a wide range of featuresif you are looking for a powerful and flexible video streaming solution, IBM Cloud Video Streaming is a great option. It offers a wide range of features and benefits that can help you to reach a global audience, deliver high-quality video, engage your audience, and secure and reliable video delivery.