KYC(Passport) API Documentation

Problem Statement:

Make An API such that:

1. User registration
2. Take an image from the user of an identity card and verify user details.

Features:

- Used mongodb as database
- Djongo + django + restframwork + python
- Extract info from id and validate data from user provided details and verify user.

Model use:

There are three model:

1. User Model : to store user details

```
class User(models.Model):
    contactNo = models.BigIntegerField(primary_key=True)
    firstName = models.CharField(max_length=50, blank=False)
    middleName = models.CharField(max_length=50, blank=True)
    lastName = models.CharField(max_length=50, blank=False)
    dob = models.DateField()
    gender = models.CharField(max_length=6)
    houseNo = models.CharField(max_length=10)
    streetName = models.CharField(max_length=50)
    localityName = models.CharField(max_length=50)
    cityName = models.CharField(max_length=20)
    countryName = models.CharField(max_length=20)
    pinCode = models.IntegerField()
    userPhoto = models.ImageField(upload_to='UserPhoto', blank=True)
```

2.IdentityCards model : to store identity card image and its type .

```
#to store id card image and its type
class IdentityCards(models.Model):
```

```
    contactNo = models.ForeignKey(User, on_delete=models.CASCADE)
    identity_type = models.CharField(max_length=100)
    identity_card_image = models.ImageField(upload_to='Passport_img',blank
=False)
      class Meta:
        unique_together = ('contactNo', 'identity_type')
```

3. Passport Details Model : to store extracted information of passport identity of a particular user.

```
#store passport details
class PassportDetails(models.Model):
    contactNo = models.ForeignKey(User, on_delete=models.CASCADE)
    first_name_on_id = models.CharField(max_length=50)
    last_name_on_id = models.CharField(max_length=50)
    gender_on_id = models.CharField(max_length=6)
    dob_on_id = models.DateField()
    country_on_id = models.CharField(max_length=20)
    id_number = models.CharField(max_length=50)
    KYC_verified = models.BooleanField(default=False)
```

Logic/design:

- Used contact number as primary key to have integrity between models
- Users can have multiple identity cards stored, and each type of card is stored once.
- First, users do the registration.
    - if successfully registered output will be user registered= true, else
    - Output the required field and errors

- Second, users upload identity card image
    - if successfully uploaded, image is processed with ai model , and is updated with the cropped image of id card,
    - Then it is processed and information is extracted using passporteye (pytesseract based ocr using mrz(machine readable zone) on passport image.
    - final output will be extracted image information, and fields that match with user details are verified.

API Design:

APIs for user registration and extraction of id card image details .
All requests are made to :http://localhost:8000

- User registration
  - Method : POST
  - URL : http://127.0.0.1:8000/api/userRegistration
  - Description:
    - This Api takes user details as input, add user in the database.
  - BODY(JSON format)

```
{
        "contactNo": 100,
        "firstName": "JODIE PIPPA",
        "middleName": "",
        "lastName": "UNITED KINGDOM FIVE",
        "dob": {
          "$date": "1985-01-17T00:00:00.000Z"
        },
        "gender": "F",
        "houseNo": "44",
        "streetName": "street",
        "localityName": "locality",
        "cityName": "city",
        "countryName": "GBR",
        "pinCode": 123456,
        "userPhoto": "face.jpg"
}
```

  - Response (JSON format)

```
{
        "user_registered": TRUE
}
```

- User verification
  - Method : POST
  - URL : http://127.0.0.1:8000/api/userPassportVerification/100/
  - Description:
    - This Api takes identity card image , and id type as input,
    - add in the database.and
    - extract information from image and add it database in separate model (passport)

- BODY(JSON format)

```
{
        "identity_type": “Passport”,
        "identity_card_image": "Passportimage.jpg"
}
```

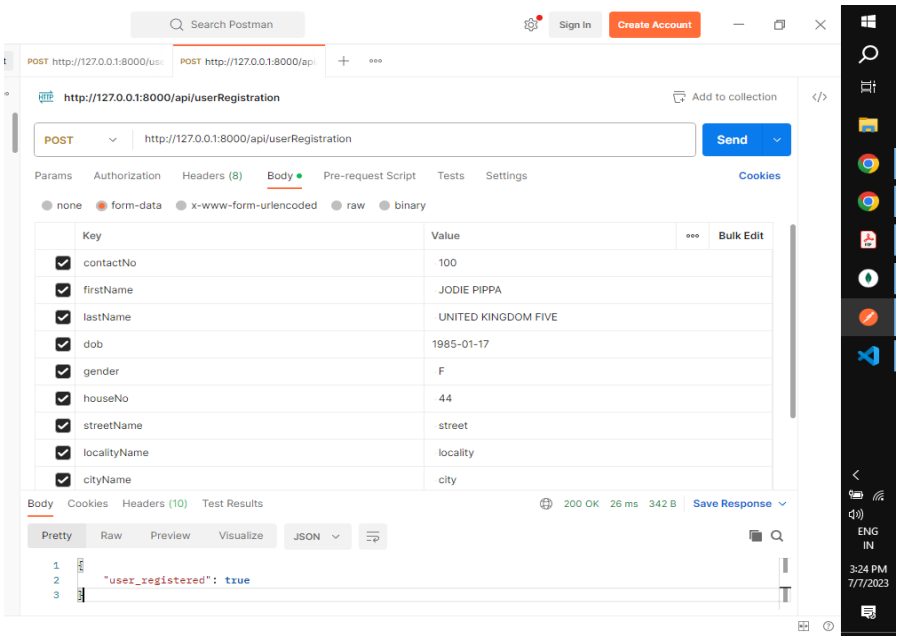
- Response (JSON format)

```
{
        "verification_status": {
            "first_name_verified": true,
            "last_Name_verified": false,
            "dob_verified": true,
            "gender_verified": true,
            "country_verified": true
        },
        "info_extrated": {
            "first_name_on_id": "JODIE PIPPA",
            "last_name_on_id": "UNITED KINGDON FIVE",
            "dob_on_id": "1985-01-17",
            "gender_on_id": "F",
            "id_number": "107185703",
            "country_on_id": "GBR"
        }
}
```

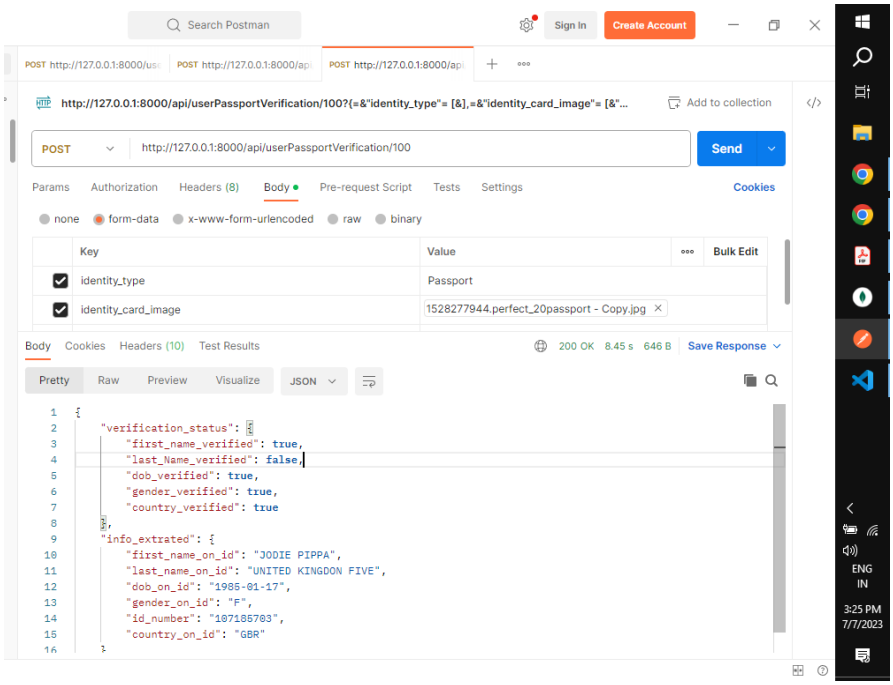
Output:

Extracted information from id and all the verified fields are set true and non verified as false.

Example:
User registration



User verification : uploads image and type , output : details and verified fields



Conclusion:

- Users can register themselves, and upload images of their id cards of various types.
- Only one image is allowed for Each type of card.
- Api is throws error if more than one image of a particular type is requested, example for passport only one image is allowed.
- This api uses pre trained ai model , which takes processing time, is accurate,
- Extraction of information is done through passport eye module which reads mrz code from passport image and output the data using pytesseract ocr.
- This api can be modified for other ids by adding different extraction function of image.

Errors

The following error codes may be returned by the API:

- 400 Bad Request: Invalid request parameters.
- 401 Unauthorized: Missing or invalid API key.
- 404 Not Found: The requested resource was not found.
- 500 Internal Server Error: An unexpected error occurred on the server.