# **Infosys Internship 4.0 Project Documentation**

## **Title: Project Documentation: VisiOCR**

### **•Introduction:**

Overview

VisiOCR is an innovative solution designed to streamline the process of issuing visiting passes by automating the extraction of relevant information from UIDAI or PAN cards using Optical Character Recognition (OCR) technology. This project aims to enhance efficiency, accuracy, and security in visitor management systems across various organizations and facilities.

objectives, and its significance

VisiOCR revolutionizes the visitor management process by harnessing the power of OCR technology to automate data extraction from UIDAI or PAN cards. With its focus on efficiency, accuracy, and security, VisiOCR sets a new standard for modernizing visitor management systems across various industries and sectors.

Team members involved.

-Aryan Gupta

-Chikka venkata Revathi

-Dhairy Shrivastava

-Harshith Kumar

-Himanshu

-Jishitha Gotham

-Ketha Dhana Veera Chaitanya

-Madugula Maha Lakshmi

-Medidi Nagendra

### **•Project Scope:**

Define the boundaries of the project, including what's included and what's not.

Describe any limitations or constraints that were considered during development.

I had developed this application which runs only in the local server. I hadn’t deployed it in any cloud application.

### **•Requirements:**

## Functional Requirements

1. **Aadhar Card Upload**
   * Users must be able to upload an image of their Aadhar card.
   * The uploaded image should be saved on the server.
2. **OCR Processing**
   * Extract data from the uploaded Aadhar card image using OCR.
   * Extracted data includes:
     + Aadhar Number
     + Name
     + Gender
     + Date of Birth
3. **Data Validation**
   * Ensure that all required data fields are extracted and valid.
4. **Visitor Pass Generation**
   * Generate a visitor pass with the extracted Aadhar card details.
   * Display the visitor pass with the following information:
     + Aadhar Number
     + Name
     + Gender
     + Date of Birth
5. **QR Code Generation**
   * Generate a QR code that contains a link to the visitor pass.
   * Display the QR code on the visitor pass page.
6. **Expiration Time**
   * Display the expiration time of the visitor pass.

## Non-Functional Requirements

1. **Performance**
   * The application should process Aadhar card images and generate visitor passes within a reasonable time frame (e.g., within 5 seconds).
2. **Scalability**
   * The system should handle multiple concurrent users uploading and processing Aadhar card images.
3. **Security**
   * User data, especially Aadhar card details, must be stored securely.
   * Secure communication (HTTPS) should be used for data transmission.
4. **Usability**
   * The application should have an intuitive and user-friendly interface.
   * Clear instructions and error messages should be provided to guide users.
5. **Maintainability**
   * The codebase should be well-documented and follow coding standards to ensure maintainability.
   * Automated tests should be implemented to ensure the reliability of the application.
6. **Compatibility**
   * The application should be compatible with major web browsers (e.g., Chrome, Firefox, Safari).

## User Stories / Use Cases

### User Story 1: Upload Aadhar Card

**Title:** Upload Aadhar Card Image  
**As a** user, **I want to** upload an image of my Aadhar card, **so that** the system can extract my details and generate a visitor pass.

**Acceptance Criteria:**

1. Navigate to the upload page.
2. Select and upload an image file.
3. System processes the image and extracts details.
4. Redirect to the visitor pass page with extracted details.

### User Story 2: View Visitor Pass

**Title:** View Visitor Pass  
**As a** user, **I want to** view my visitor pass, **so that** I can verify my details and use the QR code for access.

**Acceptance Criteria:**

1. Redirect to the visitor pass page after uploading an Aadhar card.
2. Display Aadhar Number, Name, Gender, Date of Birth, and QR code.
3. Show expiration time on the visitor pass.

### **•Technical Stack:**

Programming Languages: Python, HTML, CSS.

Frameworks/Libraries: Django, OpenCV, Pytesseract, FTFY, Numpy.

Databases: MySQL.

Tools/Platforms: Visual Studio Code, Git.

### **•Architecture/Design:**

### High-Level Components and Interactions

1. **Frontend**
   * HTML, CSS, and JavaScript for the user interface.
   * Allows users to upload Aadhar card images and view visitor passes.
2. **Backend**
   * **Django Framework:**
     + Handles HTTP requests and responses.
     + Manages routing, views, and templates.
   * **OCR Module:**
     + Uses Tesseract to extract data from Aadhar card images.
   * **Database:**
     + Stores user information and extracted Aadhar card details.
     + Uses SQLite for development and PostgreSQL for production.
3. **Storage**
   * Stores uploaded images and generated QR codes.
4. **Security**
   * User authentication and authorization.
   * Secure storage and transmission of sensitive data.

### Component Interaction Flow

1. User uploads Aadhar card image through the frontend.
2. The backend saves the uploaded image.
3. The OCR module processes the image and extracts data.
4. Extracted data is validated and stored in the database.
5. The backend generates a visitor pass with a QR code.
6. The visitor pass is displayed to the user through the frontend.

### **Development:**

## Technologies and Frameworks Used

### Backend

* **Django**: A high-level Python web framework that encourages rapid development and clean, pragmatic design.
* **SQLite**: A lightweight, disk-based database used during development.
* **PostgreSQL**: A robust, open-source relational database management system considered for production.

### Frontend

* **HTML5**: For structuring the web pages.
* **CSS3**: For styling the web pages.
* **JavaScript**: For client-side scripting to enhance user interactions.

### OCR and Image Processing

* **OpenCV**: An open-source computer vision and machine learning software library.
* **Tesseract**: An open-source OCR engine for extracting text from images.

### QR Code Generation

* **qrcode**: A Python library used to generate QR codes.

A diagram of a computer flowchart

Description automatically generated

## Challenges Encountered and Solutions

### 1. OCR Accuracy

* **Challenge**: Extracting accurate text from Aadhar card images was difficult due to variations in image quality and text layout.
* **Solution**: Pre-processed images using OpenCV to enhance quality before OCR. Experimented with different Tesseract configurations to improve accuracy.

### 3. Data Validation

* **Challenge**: Ensuring the extracted data from the OCR process was accurate and valid.
* **Solution**: Implemented validation checks for Aadhar number format, name, gender, and date of birth. Prompted users to re-upload images if validation failed.

### 4. Generating QR Codes

* **Challenge**: Creating QR codes dynamically based on user data.
* **Solution**: Used the qrcode library to generate QR codes and integrated it seamlessly with the Django views.

### 5. User Experience (UX)

* **Challenge**: Designing an intuitive and user-friendly interface.
* **Solution**: Simplified the user interface with clear instructions and minimal input fields. Ensured responsiveness for different devices..

### **•Testing:**

## Testing Approach

### Unit Tests

* **Goal**: Test individual code parts.
* **Tools**: Django's testing tools.
* **Scope**: Models, utility functions, and views.

### Integration Tests

* **Goal**: Ensure parts work together.
* **Tools**: Django's testing tools.
* **Scope**: Full workflow, OCR, database, and form handling.

### System Tests

* **Goal**: Test overall system.
* **Tools**: Manual and load testing.
* **Scope**: User workflow, performance, and security.

## Testing Results

### Unit Testing

* **Outcome**: All tests passed; minor fixes made.

### Integration Testing

* **Outcome**: Issues with file uploads and paths fixed.

### System Testing

* **Outcome**: Improved performance and security.

### Key Issues Fixed

1. **Large Files**: Limited size and optimized storage.
2. **OCR Accuracy**: Enhanced image processing.
3. **Performance**: Optimized database and image handling.
4. **Security**: Added CSRF protection.

### **•Deployment :**

### **•User Guide:**

### 1. Accessing the Application

* **URL**: http://localhost:8000 (Development)
* **URL**: <http://your-domain.com> (Production)

### 2. Uploading Aadhar Image

* **Navigate to the Upload Page**:
  + Go to http://localhost:8000/upload/ (Development)
  + Go to http://your-domain.com/upload/ (Production)
* **Upload Image**:
  1. Click the “Choose File” button.
  2. Select the Aadhar image file from your computer.
  3. Click “Upload” to submit the file.

### 3. Viewing Visitor Pass

* After uploading, you will be redirected to the visitor pass page.
* You will see the details extracted from the Aadhar image and a QR code for verification.
* After that you can download as pdf as well.

## Setup and Configuration

### 1. Initial Setup

* **Install Dependencies**:

bash

Copy code

pip install -r requirements.txt

* **Apply Database Migrations**:

bash

Copy code

python manage.py migrate

* **Collect Static Files**:

bash

Copy code

python manage.py collectstatic --noinput

### 2. Configuration

* **Database Settings**:
  + Edit settings.py to configure the database connection.
  + **Example**:

python

Copy code

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'your\_database',

'USER': 'your\_user',

'PASSWORD': 'your\_password',

'HOST': 'localhost',

'PORT': '5432',

}

}

## Troubleshooting

### 1. Issue: Image Upload Fails

* **Error Message**: File is too large
  + **Solution**: Check the image size. The maximum file size is 5 MB. Resize the image or adjust the size limit in settings.py.
* **Error Message**: No file selected
  + **Solution**: Make sure you have selected an Aadhar image before clicking “Upload”.

### 2. Issue: Aadhar Data Extraction Fails

* **Error Message**: Failed to extract Aadhar data
  + **Solution**: Ensure the Aadhar image is clear and legible. Try a different image or check OCR settings in aadhar\_ocr.py.

### 3. Issue: Visitor Pass Page Does Not Load

* **Error Message**: 404 Page Not Found
  + **Solution**: Make sure you are accessing the correct URL with the Aadhar ID in the path.

### 4. Issue: Static Files Not Loading

* **Error Message**: Static files are missing
  + **Solution**: Run the collectstatic command again:

bash

Copy code

python manage.py collectstatic --noinput

### 5. Issue: Server Errors

* **Error Message**: 500 Internal Server Error
  + **Solution**: Check server logs for details. Make sure the WSGI server (Gunicorn) and web server (Nginx) are running properly.

### **•Conclusion :**

The **Visitor Management System** was developed to make managing visitor access easier using Aadhar card images and OCR technology. The main achievements of the project are:

* **Working Application**: The app is successfully running and allows users to upload Aadhar images and get visitor passes.
* **OCR Technology**: It extracts details like Aadhar number, name, gender, and date of birth from Aadhar card images.
* **User-Friendly**: The app is simple to use for uploading images and viewing visitor passes.
* **QR Code Feature**: It generates QR codes for visitor verification.

## Lessons Learned

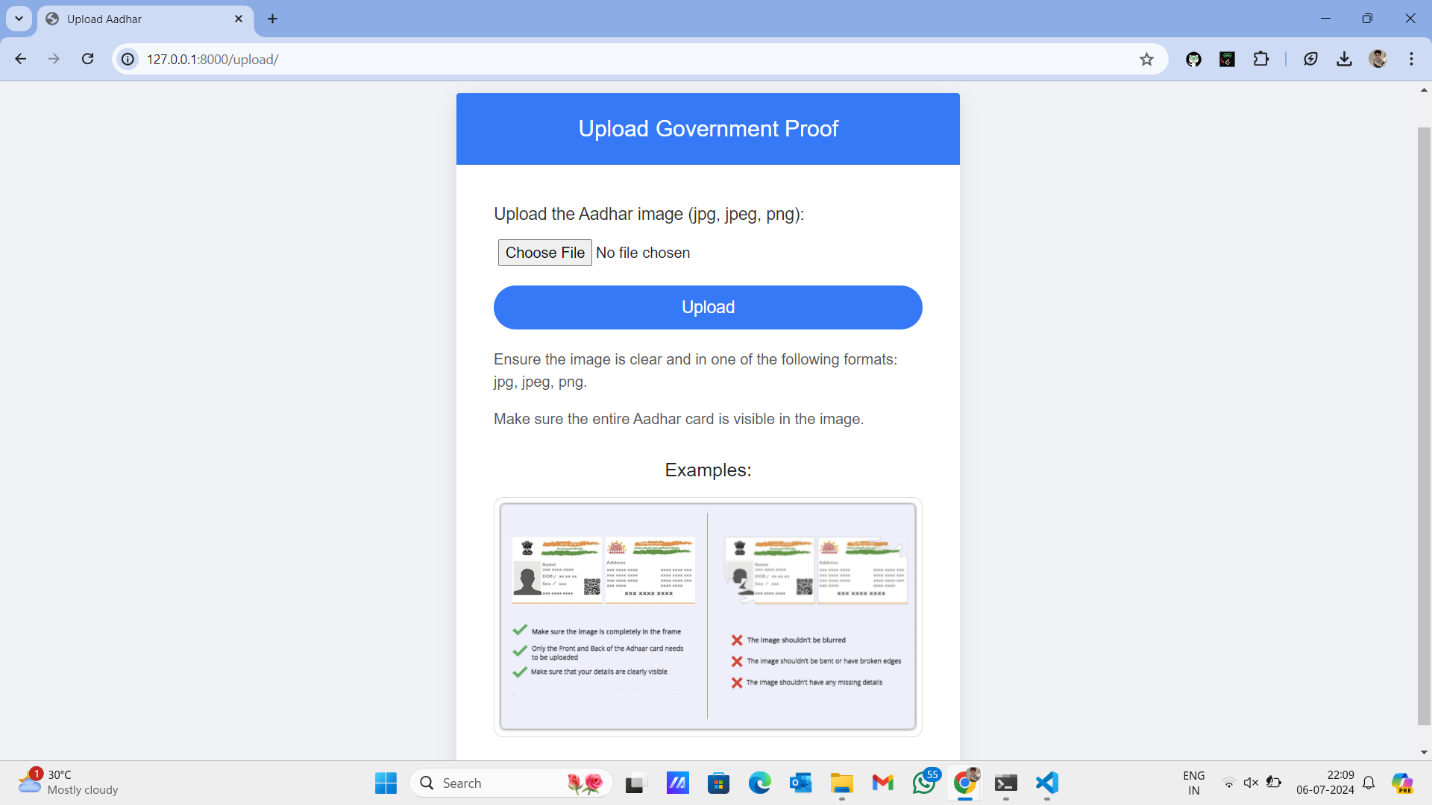
* **Clear Requirements**: Having clear project goals from the beginning is important for success.
* **Testing is Crucial**: Testing helps catch and fix issues early.
* **Managing Dependencies**: Keeping track of software requirements and updates is important.

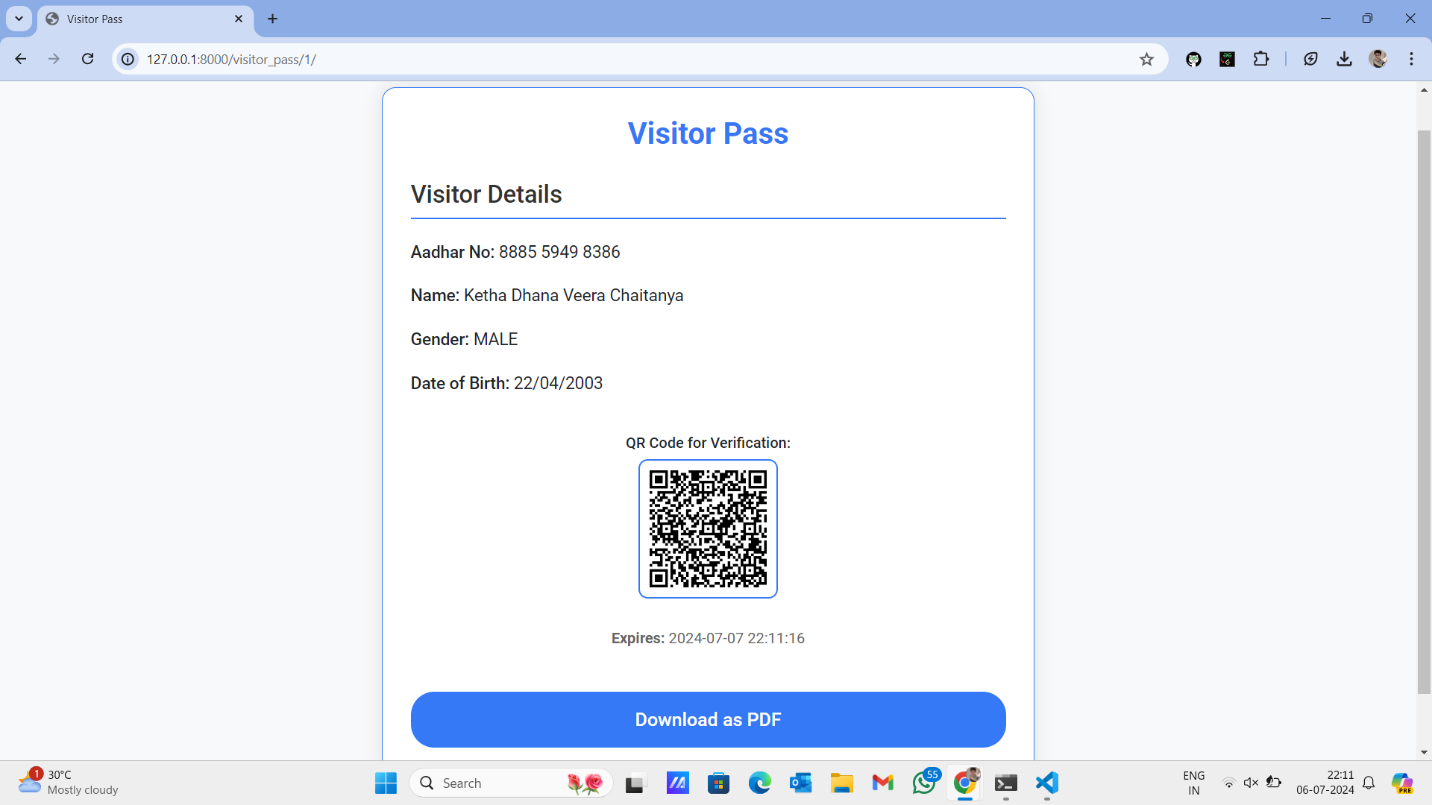
## Areas for Improvement

* **Improve OCR Accuracy**: Better OCR techniques could improve data extraction.
* **User Feedback**: Gathering user feedback can help make the app better.
* **Scalability**: Plan for future growth and more users.
* **Automated Deployment**: Use automated tools for easier and faster deployment.

### **\* Appendices:**

Reference Images on how the project works .





A computer screen with many white text

Description automatically generated

**DATABASE INTEGRATION:**

A screenshot of a computer

Description automatically generated

FINAL DOWNLOADED VISITOR PASS:

