DEPARTMENT OF COMPUTER SCIENCE ENGINEERING &
INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

Project Title - VerbalEx
Automating User Data Extraction from Government- Issued ID Cards Using OCR

MINOR PROJECT - 1,

Enrolment Numbers - 22103046, 22103048, 22103098

Name of Students - Rishika Aggarwal, Mehul Bansal, Teesha Kakkar

Name of Supervisor - Dr. Aastha Maheshwari

TABLE OF CONTENTS

Students’ Self Declaration for Open Source libraries and other source code usage in Minor Project

We **Rishika Aggarwal, Mehul Bansal, Teesha Kakkar** hereby declare the following usage of the open source code and prebuilt libraries in our minor project in **5th** Semester with the consent of our supervisor. We also measure the similarity percentage of pre written source code and our source code and the same is mentioned below. This measurement is true with the best of our knowledge and abilities.

1. List of pre build libraries :- **Tesseract, JsonWebToken**
2. List of pre build features in libraries or in source code :- **Text Extraction**
3. Percentage of pre written source code and source written by us :-

Prewritten code:- 20%
Written by us:- 80%

Student ID	Student Name	Student Signature
22103046	Rishika Aggarwal	
22103048	Mehul Bansal	
22103098	Teesha Kakkar	

Declaration by Supervisor (To be filled by Supervisor only)

I, **Dr. Aastha Maheshwari,** declares that the above submitted project with title **VerbalEx** was conducted under my supervision. The project is original and neither the project was copied from External sources nor it was submitted earlier in JIIT. I authenticate this project.

(Any Remarks by Supervisor)

Signature (Supervisor)

ACKNOWLEDGEMENT

We would like to place on record our deep sense of gratitude to **Dr. Aastha Maheshwari, ASSISTANT PROFESSOR (SENIOR GRADE)** Jaypee Institute of Information Technology, Noida for his generous guidance.

We also wish to extend our thanks to our group members and other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Signatures of the Students:

Student ID	Student Name	Student Signature
22103046	Rishika Aggarwal	
22103048	Mehul Bansal	
22103098	Teesha Kakkar	

SUMMARY

The project focuses on using Optical Character Recognition (OCR) to collect and store user data by extracting text from ID card images. This solution is aimed at creating a streamlined process for applications that require extensive data entry, multilingual support, and digital signature functionalities.

VerbalEx is a tool designed to simplify and automate user data extraction from government-issued ID cards. By capturing an image of an ID card, VerbalEx extracts essential information and organises it in a structured format.

This data can be accessed via an API, enabling various applications such as automated form filling, language translation of regional documents, and integration with e-signature services.

The extracted data can be used for pre-filling forms in different applications, saving time and minimising errors in manual data entry.Enabling translation of extracted data into regional languages, facilitating applications across multilingual audiences.

CHAPTER 1

INTRODUCTION

1.1 General Introduction:-

Text extraction from image-based documents is a transformative process that bridges the gap between non-editable visual content and structured, machine-readable data. With the proliferation of digital and scanned records across industries, efficient text extraction has become an essential tool for enhancing data accessibility and usability. This technology holds immense potential in enabling streamlined document digitization, faster content retrieval, and automation of labour-intensive tasks like data entry.

However, the challenges of text extraction cannot be overlooked. Variations in text size, font types, noise levels, and image quality make the process complex. Traditional OCR (Optical Character Recognition) systems often falter when faced with such complexities, emphasising the need for advanced methodologies that leverage modern techniques in image processing and machine learning. Addressing these challenges opens avenues for accurate and reliable extraction across diverse document types, including government-issued IDs, contracts, and personal records.

VerbalEx aims to tackle these challenges by implementing a robust, API-based solution for text extraction from government-issued ID cards. The system combines advanced OCR algorithms with meticulous pre-processing techniques to ensure high accuracy, even in the face of suboptimal image conditions. By enabling seamless integration into existing workflows, VerbalEx transforms static, image-based data into actionable insights, empowering industries to manage and utilise their information efficiently.

1.2 Problem Statement:-

In today’s digital age, a significant amount of critical data still exists in non-digital or semi-digital forms, such as scanned documents, handwritten notes, or images of text. Extracting this data accurately and efficiently poses numerous challenges, particularly when dealing with documents that exhibit variations in text quality, layout, and formatting. These challenges are magnified in specific use cases, such as processing government-issued ID cards, where precision is paramount due to the sensitive nature of the data involved.

Key Challenges:-

1. **Low Image Quality**:
   Many images of documents are captured under poor lighting conditions or with

suboptimal equipment, resulting in low resolution, noise, blurriness, or distortions. These issues significantly hinder traditional OCR systems, which struggle to interpret such degraded inputs.

2. **Text Variability**:
   Government ID cards and similar documents often feature text in multiple fonts, sizes, and orientations. Additionally, they may include a mix of printed and handwritten text, logos, seals, and watermarks, making it difficult to isolate and extract the required information accurately.
3. **Language Diversity**:
   In countries like India, documents may contain text in multiple regional languages alongside English, increasing the complexity of text recognition. Many traditional systems are not optimised to handle such multilingual content effectively.
4. **Data Structure Complexity**:
   The information on ID cards is often unstructured, requiring specialised techniques to identify and extract specific fields like names, dates, and ID numbers. Moreover, these fields may appear in inconsistent positions across different document formats.
5. **Security and Privacy Concerns**:
   Handling sensitive personal data requires robust mechanisms to ensure data privacy and prevent misuse. This adds an additional layer of complexity to any text extraction system.

1.3 Significance/ Novelty of the Problem

The significance and novelty of the problem addressed by **VerbalEx** lies in the evolving need to extract text from image-based documents, especially those containing critical information like government-issued IDs, contracts, and personal records. Here are the key points that highlight the importance and uniqueness of this problem:

1. High Demand for Document Digitization and Automation

With increasing reliance on digital systems, industries across sectors are transitioning from paper-based records to electronic formats. This creates a massive volume of image-based documents (e.g., scanned records, digital images of IDs, contracts, and forms) that need to be converted into machine-readable data for efficient management, access, and analysis. Traditional manual data entry is slow, error-prone, and costly, making automation of text extraction crucial for improving productivity and accuracy.

2. Challenges in Text Extraction from Complex, Real-World Images

Text extraction from images is far more complicated than simple scanning due to various real-world factors:

- **Text Quality and Variability**: Text size, font styles, and formatting can vary significantly, making it difficult for conventional OCR systems to extract data accurately.
- **Image Quality Issues**: Blurry, distorted, or noisy images can further complicate the extraction process.
- **Text in Complex Layouts**: Many documents, particularly government-issued IDs, have a complex layout with mixed fonts, varying sizes, and irregular spacing that traditional OCR often struggles with.

Addressing these issues requires a sophisticated solution that not only reads the text but also adapts to the wide variety of document types, ensuring accuracy under challenging conditions.

3. Current Limitations of Traditional OCR Systems

Most traditional OCR systems fall short when dealing with variations in image quality, complex document layouts, and noise, often leading to:

- **Inaccurate Extraction**: Misread characters, incorrect formatting, and incomplete extraction of data.
- **Limited Usability**: The output data may require significant post-processing and manual intervention, negating the benefits of automation.

These shortcomings highlight the need for more advanced methodologies that combine OCR with image pre-processing techniques and machine learning to enhance accuracy.

4. VerbalEx's Novelty in Addressing the Problem

- **Advanced OCR Algorithms**: By combining state-of-the-art OCR with sophisticated image pre-processing techniques, VerbalEx ensures more accurate extraction from even noisy, low-quality images.
- **Seamless API Integration**: VerbalEx offers an API-based solution, allowing businesses to integrate the text extraction capability into their existing workflows without major disruption or infrastructure overhaul.
- **Focus on Government-Issued IDs and Sensitive Documents**: By specialising in extracting data from government IDs, VerbalEx focuses on a high-value use case where accuracy, security, and reliability are paramount. This is particularly relevant for industries like banking, insurance, healthcare, and government services that deal with sensitive data.
- **Real-World Adaptability**: VerbalEx's ability to handle a range of image conditions and document types, from high-quality scans to low-res photos, positions it as a versatile and reliable solution for businesses and organisations.

5. Enabling Efficiency and Accessibility

- **Faster Content Retrieval**: By converting static, image-based content into structured, machine-readable text, VerbalEx enables faster data retrieval and access, improving operational efficiency.
- **Improved Data Accessibility**: The structured format allows for easier searching, processing, and analysis of data, unlocking the potential for enhanced decision-making and insights.
- **Empowering Automation**: With reliable text extraction, businesses can automate tasks that were once time-consuming, such as data entry, verification, and document processing, leading to time and cost savings.

1.4 :-Empirical Study:-

An empirical study was conducted to analyse the effectiveness of current tools and technologies available for text extraction from image-based documents, particularly focusing on government-issued ID cards. This survey was carried out through a combination of field surveys, a review of existing tools, and experimentation with leading OCR technologies, aimed at identifying the strengths, weaknesses, and limitations of current solutions.

Existing Tools Survey:-

The existing tools for text extraction from image-based documents were thoroughly analysed to understand their capabilities and limitations. The key tools explored include traditional OCR systems, machine learning-based approaches, and specialised tools for document analysis. This survey aimed to assess the performance of these tools in extracting text from complex document layouts like government-issued IDs.

1. Tesseract OCR

Tesseract is an open-source OCR library widely used across various industries for text extraction. It excels in multilingual text recognition, making it versatile for many use cases. However, Tesseract struggles with complex layouts, mixed fonts, irregular spacing, and low-quality images, often leading to inaccuracies when processing government-issued IDs under suboptimal conditions.

2. Google Cloud Vision OCR

Google Cloud Vision OCR leverages deep learning algorithms to extract text from images and performs well with high-quality input. While it handles clear images effectively, it faces limitations with poor-quality images, non-Latin scripts, and documents with intricate layouts like those found on government IDs. Additionally, its paid model can be a barrier for cost-sensitive organisations.

3. ABBYY FineReader

ABBYY FineReader offers high accuracy and advanced document recognition, especially for scanned PDFs and professional-grade documents. It excels in well-preserved documents but struggles with images captured under less-than-ideal conditions, such as smartphone photos. The tool's high cost also makes it difficult to scale for large, budget-constrained projects that require widespread deployment.

4. Adobe Acrobat Pro

Adobe Acrobat Pro is a well-established tool for document management with OCR functionality for scanned PDFs. Despite its usefulness, it encounters issues with diverse fonts, layouts, and languages, particularly for image-based documents like photographs of government IDs. It is more suitable for scanned PDFs than images from non-traditional sources, limiting its applicability in this context.

Field Survey:-

A field survey was conducted to gather insights from potential users and stakeholders in industries such as banking, insurance, healthcare, and government. The focus was on understanding challenges with manual data entry, especially when dealing with image-based documents like government-issued IDs. Respondents shared their experiences with existing tools and their expectations for an ideal text extraction solution.

Key findings include:

- **Accuracy:** High accuracy was emphasised, especially for extracting sensitive personal information.
- **Support for diverse formats and languages:** There was a need for multilingual support, especially for languages used in countries like India.
- **Ease of integration:** Users wanted a solution that could easily fit into existing workflows.
- **Data privacy concerns:** Ensuring secure handling of sensitive data was a top priority.

Experimental Survey:-

An experimental survey was conducted to test the tools with sample images of government-issued IDs under various conditions (e.g., varying image quality, fonts, languages). The goal was to evaluate the real-world performance of each tool.

Key findings revealed:

- **Low-quality image handling:** Tesseract and Google Cloud Vision OCR struggled with low-resolution or noisy images.
- **Multilingual extraction:** Tools often failed with non-Latin scripts, such as Hindi or Bengali.
- **Layout complexity:** Documents with logos, seals, or mixed fonts were challenging for many tools.

1.5 Brief description of our solution Approach:-

Our solution approach revolves around leveraging the Tesseract OCR library for accurately extracting text from images. Tesseract, known for its robust multilingual text recognition capabilities, allows us to handle complex document layouts and diverse languages effectively. The extracted text is then processed and stored in MongoDB, enabling efficient management and retrieval of digitised data. This pipeline ensures that even large volumes of unstructured information can be transformed into a structured and accessible format.

To provide a seamless user experience, we integrate the OCR functionality into a user-friendly web platform. Built with React and Node.js, the platform allows users to upload document images and view the extracted data in real-time. MongoDB acts as the backbone for storing this data, enabling scalable storage and quick access for future use. This approach ensures that our solution is not only reliable but also scalable, capable of handling various document types such as government IDs, forms, or handwritten notes.

By automating the data extraction process and securely storing the results in a database, our solution bridges the gap between physical documents and digital systems. This efficient and scalable approach addresses the challenges of digitising diverse document formats while providing a robust foundation for future enhancements like data analytics or API integrations.

In conclusion, our project effectively addresses the challenges of digitising and managing unstructured data from physical documents by utilising the Tesseract OCR library for text extraction and MongoDB for secure and scalable data storage. This combination of robust OCR capabilities and a reliable database ensures that diverse document types, including complex layouts and multilingual content, can be digitised with precision and efficiency.

1.6 Comparison of Existing Approaches:-

While all tools have strengths in specific contexts, none fully address the challenges posed by government-issued ID extraction. **Tesseract** and **Google Cloud Vision OCR** show potential but struggle with low-quality images and complex document layouts. **ABBYY FineReader** and **Adobe Acrobat Pro** provide high accuracy with structured documents but fail with image-based IDs or non-ideal conditions. Therefore, an ideal solution would require a hybrid approach, integrating robust OCR capabilities with advanced pre-processing techniques to handle diverse formats, languages, and image qualities.

CHAPTER 2

Literature Survey

2.1 Summary of papers studied:-

1. **Tesseract OCR (Rath et al., 2007)**: This paper describes the development of Tesseract, one of the most widely used open-source OCR engines. It highlights its effectiveness in multilingual text recognition, but also points out its limitations in dealing with low-quality images and complex layouts, such as those often found in government-issued IDs.

2. **Google Cloud Vision OCR (Google, 2016)**: This paper explains the deep learning models behind Google Cloud Vision OCR, which is a cloud-based tool for text recognition. It performs well on high-quality images but struggles with multilingual support and images with distorted or noisy backgrounds, making it less suitable for certain real-world use cases.

3. **ABBYY FineReader (ABBYY, 2019)**: The ABBYY FineReader software offers high accuracy for OCR tasks involving scanned documents. However, it does not perform as well with images taken by smartphone cameras or images with poor resolution. The paper also discusses the tool's high licensing costs, which limits its adoption in resource-constrained environments.

4. **Deep Learning for Text Extraction (Zhang et al., 2018)**: This paper focuses on recent advancements in deep learning-based approaches for text extraction. It highlights the success of convolutional neural networks (CNNs) in improving text extraction accuracy, particularly for challenging documents with complex layouts and fonts. However, it also points to the need for better data preprocessing techniques to improve performance on noisy images.

2.2 Integrated summary of the literature studied:-

The literature on OCR and text extraction from images reveals significant progress, particularly in the areas of multilingual support, accuracy, and the handling of complex document layouts. Traditional OCR systems, such as **Tesseract**, have proven useful for a variety of text extraction tasks but face challenges with low-quality images and intricate document layouts. Meanwhile, modern tools like **Google Cloud Vision OCR** and **ABBYY FineReader** provide higher accuracy for clearer documents but struggle with non-Latin scripts, noisy images, and the high cost of licensing.

In recent years, the integration of **deep learning** models has significantly enhanced text extraction capabilities, allowing for better handling of complex documents. However, these approaches still face challenges in real-world scenarios, particularly in dealing with documents that contain distorted images, mixed fonts, or multiple languages. The literature also emphasises the importance of **image preprocessing** techniques, such as noise reduction and image enhancement, to improve OCR performance on low-quality images.

The integrated analysis suggests that a hybrid approach combining traditional OCR with modern machine learning techniques, along with effective preprocessing, holds promise for addressing the challenges identified in existing tools. This hybrid model could offer an effective solution for accurate and reliable text extraction from image-based documents, particularly for government-issued IDs that often feature mixed fonts, varied layouts, and poor image quality.

CHAPTER 3

REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1 Overall Description of the Project:-

❖ Frontend Development:-

1. **React.js**: Utilised React.js for creating a highly dynamic and responsive user interface. The component-based architecture of React allowed for the creation of reusable, modular UI elements, improving code maintainability and scalability.
2. **React Router DOM**: Implemented React Router DOM to manage the routing system of the application. This enabled seamless transitions between pages, ensuring a smooth user experience as users navigated through different sections of the app.
3. **Axios**: Integrated Axios, a promise-based HTTP client, to handle asynchronous API calls between the frontend and the backend. This ensured seamless and real-time data exchange, improving the responsiveness of the application.
4. **State Management**: Leveraged React's state management techniques (such as useState, useEffect) to efficiently manage and pass data between components. This optimised data flow and reduced unnecessary re-renders.

❖ Backend Development:

- Established an Express.js server to handle HTTP requests and responses, providing a robust backend infrastructure for the application.
- Utilised Mongoose, an object data modelling (ODM) library for MongoDB, to interact with the MongoDB database hosted on MongoDB Atlas, facilitating seamless data storage and retrieval operations.
- Implemented JSON Web Tokens (JWT) for secure authentication and authorization, ensuring that users can access protected resources and endpoints securely.
- Employed various Express middlewares such as cookie-parser and body-parser for parsing cookies and request bodies, enhancing the server's functionality and handling of incoming requests.
- Implemented robust error-handling mechanisms and standardised response formats to ensure consistency and reliability in API interactions and server responses.

- Designed and implemented RESTful API endpoints following industry best practices, adhering to conventions for resource naming, HTTP methods, and status codes to ensure scalability, maintainability, and interoperability of the backend system.
- Implemented MongoDB optimization techniques like indexing and aggregation to improve query performance, especially when dealing with large datasets and complex queries.

3.2 Requirement Analysis:-

The requirement analysis for extracting text from image-based documents, particularly government-issued IDs, highlights several key needs:

1. **Accuracy:** The solution must be able to extract information with high accuracy, especially when dealing with sensitive personal data. Errors in text extraction can lead to significant issues in industries such as banking and healthcare.
2. **Handling Low-Quality Images:** ID documents may be scanned or photographed under poor conditions. The solution should be able to handle low-resolution images and still extract text effectively.
3. **Multilingual Support:** Government-issued IDs may contain text in multiple languages, including regional scripts. The tool must be capable of recognizing and processing different languages and fonts.
4. **Complex Layouts:** Many IDs contain logos, stamps, and varied fonts. The solution should be able to process documents with these complex layouts and extract text accurately.
5. **Security:** Since IDs contain personal and sensitive information, the solution must ensure that data privacy and security are maintained throughout the extraction process.
6. **Ease of Use and Integration:** The solution should be simple to use and integrate into existing workflows without requiring major changes or new infrastructure.

3.3 Solution Approach:-

To address the identified requirements, the solution will adopt a straightforward approach:

1. **Image Pre-processing:** Before extracting text, the images will be pre-processed to improve their quality. This can include basic techniques like adjusting brightness and contrast or applying filters to reduce noise, making the text clearer for extraction.
2. **OCR Tool Selection:** A basic Optical Character Recognition (OCR) tool like **Tesseract OCR** will be used to extract text from images. Tesseract is a reliable open-source OCR engine that works well with clear images, and it supports various languages.
3. **Multilingual Support:** Tesseract supports multiple languages, so the solution will focus on ensuring the correct language model is used for the specific document. This will help accurately process regional scripts that are often found on government-issued IDs.
4. **Post-processing:** After the text is extracted, a simple validation process will check for common errors, such as missing characters or misread words, and correct them where possible.

5. **Security and Privacy:** The extracted data will be stored securely, ensuring that personal information is protected. Basic encryption methods will be applied to safeguard sensitive data.
6. **User Interface:** The solution will include a simple interface where users can upload documents and view the extracted text. This ensures the solution is easy to use, even for non-technical users, and integrates smoothly into existing workflows.

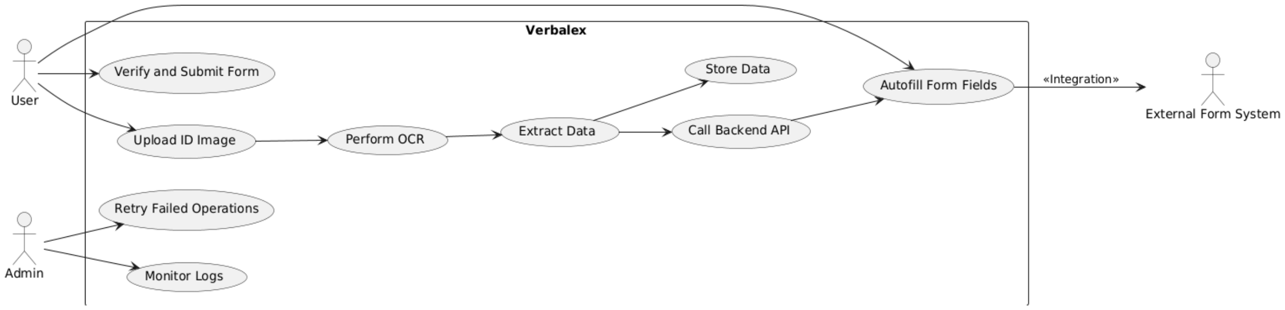
CHAPTER 4

MODELLING AND IMPLEMENTATION DETAILS

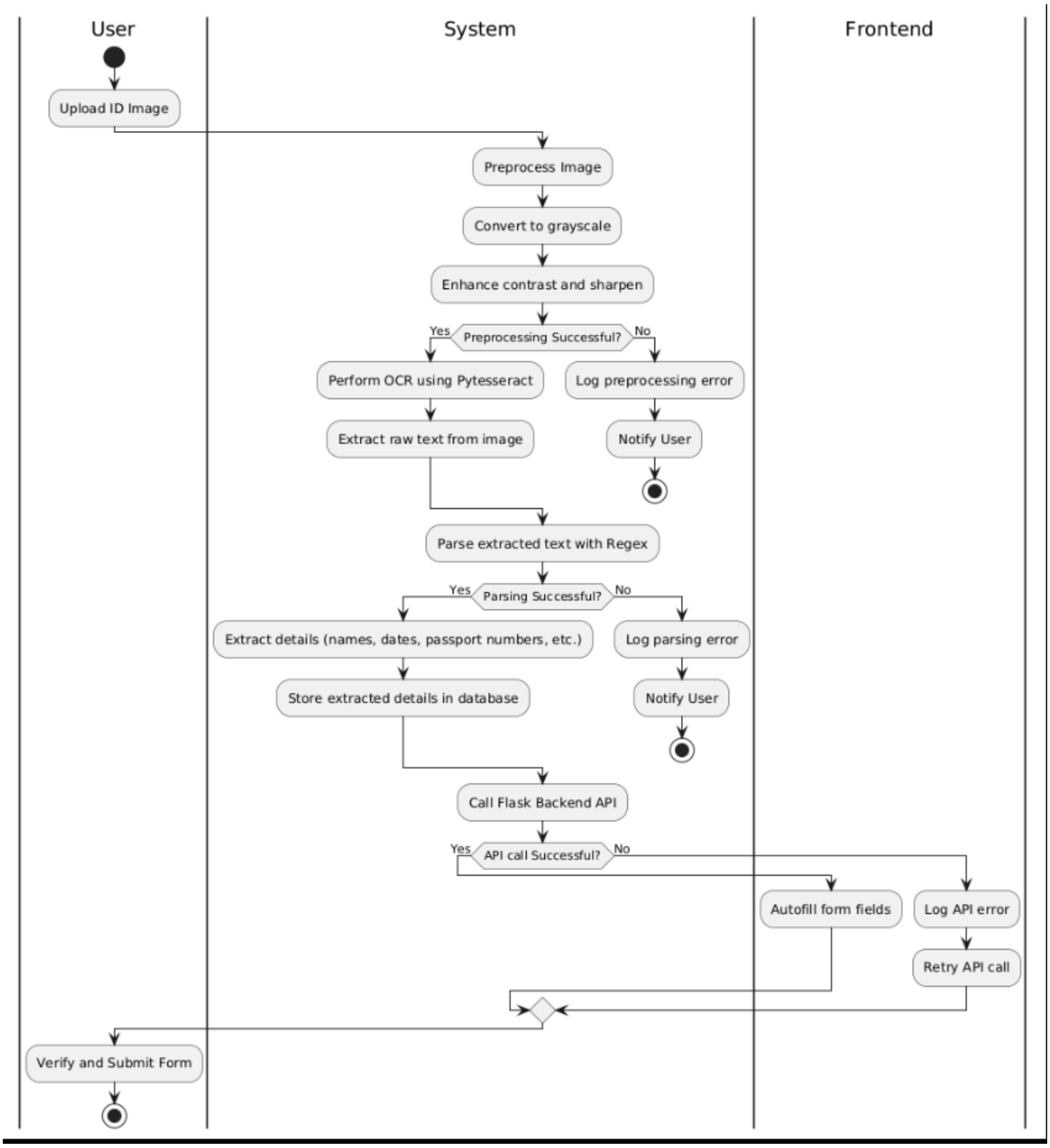
The modelling and implementation of the text extraction system for government-issued IDs involve several key stages, from understanding the problem to creating a practical solution. Below are the details of how the system is modelled and implemented:

4.1 Design Diagrams:-

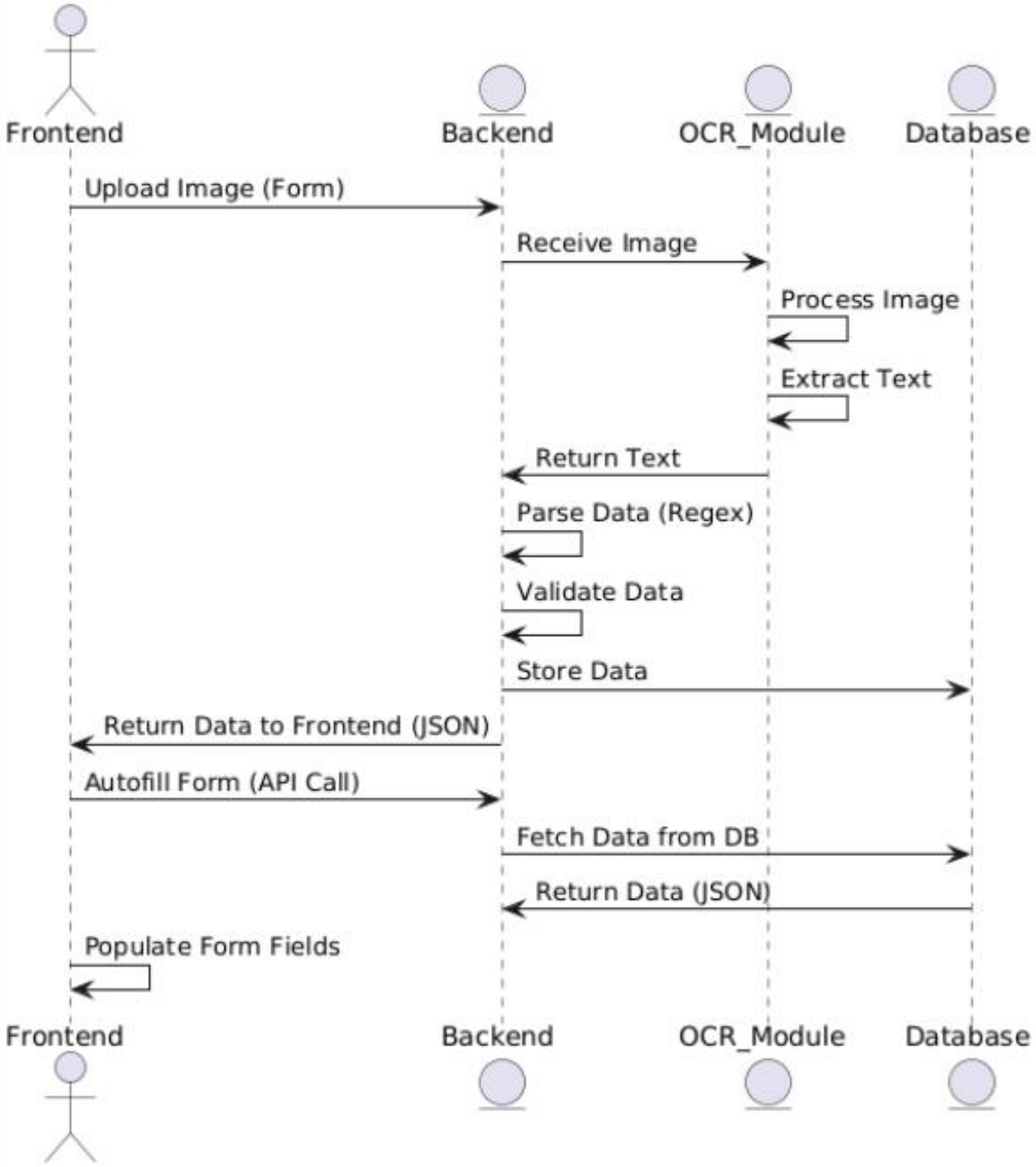
4.1.1 Use case diagram:-



4.1.2 Control flow diagram:-



4.1.3 Sequence diagram:-



4.2 Implementation Details and Issues:-

❖ Frontend Implementation Details :-

1) Component-Based Layouts using React.js:

● Developed modular UI components using React.js, such as Navbar, Footer, Profile Cards, Forms, etc., to create a structured and intuitive user interface.
● Organised components into a hierarchy to manage complex layouts efficiently. Stateful Components:
● Utilised React class components with hooks to manage component state and lifecycle events.

2) React-Router-DOM Integration:

● Defined routes using React Router DOM to handle navigation between different pages and views within the application.
● Set up route components for each page or feature, ensuring proper rendering and management of UI states.

3) API Calls with Axios:

● Used Axios, a promise-based HTTP client, to make asynchronous API calls to the backend server.
● Implemented functions to fetch data from API endpoints, handle responses, and update UI components with the retrieved data.

4) Form Handling and Validation:

● Implemented basic validation checks while filling forms to ensure that the user-provided data meets specific criteria (e.g., required fields, valid email format, password).

❖ Backend Implementation Details

1) Express.js Server with Mongoose for MongoDB:

● Set up an Express.js server to handle HTTP requests and responses, serving as the backend infrastructure for SkillSwap.
● Utilised Mongoose, an object data modelling (ODM) library for MongoDB, to interact with the MongoDB database hosted on MongoDB Atlas.
● Defined database schemas and models using Mongoose to structure and manage data entities, ensuring consistency and integrity in data storage.

2) JSON Web Token (JWT) for Authentication and Authorization:

● Used the JSON Web Token (JWT) package to generate, verify, and manage JWT tokens for user authentication and authorization.
● Implemented JWT-based authentication middleware to secure protected routes and endpoints, ensuring that only authenticated users can access authorised resources.

3) Express Middleware for Request Parsing:

● Incorporated Express middlewares like cookie-parser for parsing cookies, URL-encoded for handling URL queries, and other relevant middlewares for request parsing and processing.
● Configured middleware pipelines to preprocess incoming requests, extract necessary data, and perform validations or transformations as needed.

4) RESTful API Endpoints:

● Designed and implemented API endpoints following the principles of RESTful API architectural design, adhering to conventions for resource naming, HTTP methods, and status codes.
● Defined routes and controllers for CRUD operations on various resources, ensuring consistency and predictability in API behaviour.

4.3 Risk Management and Mitigation:-

In any project, especially one involving the extraction of sensitive data from image-based documents, it is crucial to identify potential risks and have strategies in place to mitigate them. For the project focused on text extraction from government-issued IDs, several risks need to be considered.

Accuracy Risk:

- **Risk:** The primary risk in OCR-based projects is the accuracy of text extraction. Poor quality images or complex layouts can result in incorrect or incomplete data extraction.
- **Mitigation:** Implement pre-processing techniques to enhance image quality, such as noise reduction and contrast adjustments. Additionally, use an iterative approach to refine OCR results and perform manual validation on critical data fields.

Legal and Compliance Risks:

- **Risk:** Extracting and processing sensitive personal data may face legal or regulatory hurdles, especially in regions with strict data protection laws (e.g., GDPR in Europe or data privacy laws in India).

- **Mitigation:** Ensure that the solution complies with local data privacy regulations and that all stakeholders are aware of legal requirements. Obtain user consent where necessary, and include mechanisms for data anonymization or deletion when required.

Security and Privacy Risks:

- **Risk:** Handling sensitive data, such as personal details on government-issued IDs, introduces security and privacy concerns. If data is not properly secured, it can lead to breaches or misuse.
- **Mitigation:** Ensure that all extracted data is encrypted both in transit and at rest. Implement role-based access control (RBAC) to restrict data access and ensure that only authorised personnel can handle sensitive information.

Integration Risk:

- **Risk:** Integrating the text extraction solution into existing systems or workflows may face compatibility issues, especially if the system is outdated or lacks necessary infrastructure.
- **Mitigation:** Design the solution to be lightweight and easily integrable with existing systems. Provide clear documentation and support to assist with integration. Conduct pilot testing with a small user group to ensure smooth integration before full deployment.

CHAPTER 5

TESTING

5.1 Testing Plan:-

The testing plan for **VerbalEx** outlines a systematic approach to ensure the quality, functionality, and performance of the system. The goal is to confirm that each component and the overall system perform as expected under various conditions. The testing plan includes multiple stages, starting from individual component validation to real-world use case simulation. Below are the key stages of testing:

❖ **Unit Testing**:

- **Purpose**: Test individual modules like image preprocessing, text extraction, and data post processing in isolation.
- **Method**: Use mock inputs to check if each component performs its task correctly.
- **Tools**: Jest, Mocha for JavaScript testing, and other testing frameworks.
- **Examples**:
    - Verify image preprocessing filters out noise and improves image quality.
    - Validate that OCR extracts text from clear images without errors.

❖ **Integration Testing**:

- **Purpose**: Ensure that components interact as expected when integrated.
- **Method**: Test data flow from one module to the next, starting from image upload to text extraction and storage.
- **Tools**: Postman for API testing, Chai for assertions in Node.js.
- **Examples**:
    - Test the entire image upload, OCR extraction, and data storage process.
    - Verify that the system integrates with the database to store extracted text.

❖ **System Testing**:

- **Purpose**: Validate that the entire system functions together in a real-world environment.
- **Method**: Simulate end-to-end scenarios, where users upload documents, extract text, and view results.
- **Tools**: Selenium for automated browser testing, manual tests.
- **Examples**:
    - Test the user experience by uploading various types of documents (clear, blurry, multilingual).
    - Verify that all error messages and system feedback are displayed correctly.

❖ **Regression Testing**:

- **Purpose**: Ensure that new updates or changes do not break existing features.
- **Method**: Retest previously successful test cases after every code update.
- **Tools**: Automated test scripts.
- **Examples**:
    - After modifying the image preprocessing algorithm, recheck that text extraction is still accurate.
    - Ensure that any UI changes don’t disrupt the user experience.

5.2 Component decomposition and type of testing required:-

The VerbalEx system can be decomposed into the following key components, each playing a crucial role in ensuring the smooth functioning of the overall solution:

1. **Image Preprocessing Module**:
    - **Function**: This module handles the preprocessing of image data to enhance the quality and prepare it for text extraction. It performs operations like noise reduction, image resizing, contrast enhancement, and binarization.
    - **Goal**: To improve the accuracy of text extraction by converting images into a format that the OCR system can process effectively.
    - **Additional Features**: It also includes the ability to detect orientation issues and perform automatic rotation to align the image properly.

2. **Text Extraction Module**:

- **Function**: This module uses Optical Character Recognition (OCR) techniques to extract text from preprocessed images. The text extraction process involves recognizing characters and converting them into machine-readable text.
- **Goal**: To convert image-based text into editable, searchable, and usable content.
- **Challenges**: The module must handle various fonts, layouts, and languages to ensure accuracy, especially when dealing with images like government IDs that contain mixed fonts or handwritten text.

3. **Data Post processing Module**:
    - **Function**: Once the text is extracted, this module refines and organises it. It removes any OCR errors, applies grammar corrections, and formats the text to make it more readable. It also handles tasks like separating key fields (e.g., names, dates of birth, addresses) from the raw extracted data.
    - **Goal**: To ensure that the text output is clean, structured, and ready for use or further analysis.
    - **Additional Features**: It includes techniques for handling multilingual text and formatting issues that may arise due to diverse document structures.

4. **User Interface (UI)**:
    - **Function**: The UI allows users to interact with the system. It provides an intuitive platform where users can upload images, view the extracted text, and download processed data.
    - **Goal**: To offer a seamless and user-friendly experience that simplifies the document extraction process for non-technical users.
    - **Additional Features**: The UI supports features such as real-time feedback on image quality, extraction progress, and downloadable output formats like TXT or CSV.

5. **Database Module**:
    - **Function**: This module is responsible for storing the extracted text along with associated metadata (e.g., document type, extraction date) for future reference, validation, or auditing.
    - **Goal**: To provide a reliable storage mechanism for both the original and processed data, ensuring easy retrieval when needed.
    - **Additional Features**: It integrates with the rest of the system to allow users to search, view, and manage previously processed documents and their metadata.

6. **Error Handling and Logging**:
   - **Function**: This component tracks and handles errors that occur during the extraction process. It logs issues, provides meaningful feedback to the user, and suggests corrective actions.
   - **Goal**: To ensure that any disruptions in the workflow are captured, and users are informed about potential issues, such as poor image quality or extraction failures.
   - **Additional Features**: Includes a logging system that tracks performance metrics, errors, and system usage, which is valuable for debugging, improvement, and future updates.

5.3 List all test cases in prescribed format:-

Test Case 1: Image Upload Test

- **Test Case Title**: Verify the image upload functionality
- **Description**: Check if users can successfully upload images in supported formats (e.g., JPG, PNG).
- **Preconditions**: User is logged into the system.
- **Test Data**: Sample image files (JPG, PNG).
- **Test Steps**:
  1. Navigate to the image upload section.
  2. Select an image file (JPG/PNG).
  3. Click on the "Upload" button.
- **Expected Result**: The image is uploaded successfully, and a preview is displayed on the screen.

Test Case 2: Image Preprocessing Test

- **Test Case Title**: Verify image preprocessing for noise reduction
- **Description**: Ensure that the image preprocessing module reduces noise and enhances image quality.
- **Preconditions**: Image is uploaded successfully.
- **Test Data**: Sample noisy image (low-resolution, distorted).
- **Test Steps**:
  1. Upload a noisy image.
  2. The system processes the image to reduce noise.
  3. Observe the output image.

- **Expected Result**: The image should appear clearer and with reduced noise, ready for OCR extraction.

Test Case 3: Text Extraction Test

- **Test Case Title**: Verify text extraction accuracy
- **Description**: Ensure that the text is accurately extracted from the preprocessed image.
- **Preconditions**: Image is preprocessed and ready for OCR.
- **Test Data**: Preprocessed image containing clear, readable text.
- **Test Steps**:
    1. Upload a clear image.
    2. System runs OCR and extracts the text.
    3. Verify the extracted text.
- **Expected Result**: The text should be accurately extracted without errors.

Test Case 4: Multilingual Text Extraction Test

- **Test Case Title**: Verify multilingual text extraction
- **Description**: Check if the system extracts text accurately from images containing multiple languages.
- **Preconditions**: Image with multilingual text is uploaded.
- **Test Data**: Sample image with Hindi and English text.
- **Test Steps**:
    1. Upload an image containing multilingual text (e.g., Hindi and English).
    2. System extracts text from the image.
    3. Verify the extracted text for both languages.
- **Expected Result**: The extracted text should accurately reflect both languages (e.g., Hindi and English) without errors.

Test Case 5: Data Post processing Test

- **Test Case Title**: Verify data post processing (text correction and formatting)
- **Description**: Ensure that extracted text is post processed, correcting any OCR errors and formatting it properly.
- **Preconditions**: OCR has extracted the text.
- **Test Data**: Sample text with common OCR errors (e.g., missing characters, spacing issues).
- **Test Steps**:
    1. Process extracted text through the data post processing module.

   2. Review the formatted and corrected text.
- **Expected Result**: The text should be free from errors and properly formatted.

Test Case 6: User Interface Test

- **Test Case Title**: Verify user interface for uploaded image preview
- **Description**: Ensure that the uploaded image preview is displayed correctly in the user interface.
- **Preconditions**: Image is uploaded.
- **Test Data**: Uploaded image file (JPG/PNG).
- **Test Steps**:
   1. Upload an image file.
   2. Verify that the uploaded image is displayed as a preview.
- **Expected Result**: The image should appear correctly on the UI, with an accurate preview.

Test Case 7: Error Handling Test

- **Test Case Title**: Verify error handling for invalid file type
- **Description**: Check if the system properly handles invalid image file formats.
- **Preconditions**: User attempts to upload an invalid file format.
- **Test Data**: Invalid file format (e.g., PDF, DOCX).
- **Test Steps**:
   1. Attempt to upload a file in an unsupported format (e.g., PDF).
   2. Observe the error message displayed.
- **Expected Result**: The system should show a clear error message indicating that the file format is unsupported.

Test Case 8: Database Storage Test

- **Test Case Title**: Verify text storage in database
- **Description**: Ensure that extracted text is correctly stored in the database.
- **Preconditions**: Text has been extracted and post processed.
- **Test Data**: Extracted text from an image.
- **Test Steps**:
   1. Extract text from an image.
   2. Store the extracted text in the database.
   3. Retrieve the stored text from the database.

- **Expected Result**: The text should be correctly stored and retrievable from the database.

5.4 Error and Exceptional Handling:-

The **VerbalEx** system includes robust error and exception handling mechanisms to ensure a seamless user experience. Some of the key error handling strategies are:

1. **Invalid Input Files**: If the user uploads a non-image file (e.g., PDF or DOCX), the system will return an error message, prompting them to upload a valid image file.
2. **Image Quality Issues**: If the uploaded image is of low quality (e.g., blurry, noisy, or distorted), the system will attempt to preprocess the image, but it may return a warning indicating that the extraction might be inaccurate. The user will be encouraged to upload a clearer image for better results.
3. **Text Extraction Errors**: If the text extraction process fails due to a complex layout or unrecognised fonts, the system will log the error and display a generic error message to the user, suggesting retrying with a different image.
4. **Multilingual Handling**: If the system detects unsupported languages or scripts, it will alert the user and attempt to extract the text in the supported language(s) if possible.

5.5 Limitations of the Solution:-

Despite its strengths, the **VerbalEx** system does have some limitations:

1. **Accuracy in Low-Quality Images**: The accuracy of text extraction is significantly reduced when the image quality is poor, such as blurry or noisy government IDs. Although preprocessing helps improve image quality, the solution may still struggle with highly degraded images.

2. **Complex Layouts and Mixed Fonts**: Documents with complex layouts (e.g., images with logos, seals, or watermarks) or mixed fonts (e.g., varying sizes and styles) may not be processed accurately, resulting in errors or incomplete extractions.

3. **Limited Language Support**: While the system supports multiple languages, it may struggle with non-Latin scripts or languages with complex characters and structures, leading to incorrect or incomplete text extraction.

4. **Limited Real-Time Processing**: The current solution may not be suitable for real-time processing of large volumes of images due to performance constraints. The time required to process large files could be a limiting factor in high-demand environments.

5. **Dependency on Image Quality**: The success of the extraction process heavily depends on the quality of the input images. In real-world use, especially with government-issued IDs, image quality can vary, and this can limit the system's effectiveness in certain scenarios.

CHAPTER 6

FINDINGS, CONCLUSION, AND FUTURE WORK

6.1 Findings:-

Upon completing our project work for **VerbalEx**, we have observed the following key outcomes:

1. **Intuitive User Interface**: The VerbalEx website offers an intuitive and user-friendly interface, with a well-organised layout that makes navigation effortless. Users can easily browse through various sections, such as profile creation, communication, and resources, ensuring smooth access to key features without any significant learning curve.
2. **Responsive Design**: The platform is designed with responsiveness in mind, ensuring that it functions seamlessly across a wide range of devices, including desktops, laptops, tablets, and smartphones. Whether users access the site from a larger screen or a mobile device, they experience consistent and optimised browsing, which is critical for modern, on-the-go accessibility.
3. **Engaging Visuals**: VerbalEx incorporates visually appealing design elements, such as high-quality images, modern icons, and custom graphics. These visuals not only enhance the overall aesthetics of the website but also improve user experience by making it more engaging and helping users to easily identify different sections and features.
4. **Secure Authentication**: The platform prioritises user privacy and security through robust authentication mechanisms. By using secure login methods, VerbalEx ensures that users' personal information and account data are protected, fostering trust and reliability in the platform.

6.2 Future Work:-

The development of **VerbalEx** has laid the foundation for a powerful OCR-based system, but there are several areas for future improvement and expansion. Below are the key areas we plan to focus on:

1. **Language Recognition**:
   - Implement support for recognizing multiple languages, allowing the system to handle documents in various regional languages. This will make the system more versatile and applicable for a broader range of users.
2. **Improvement of User Interfaces**:
   - Refine the user interfaces to make the system more intuitive and user-friendly. Enhancing the design and user experience will ensure smoother interaction with the tool.
3. **Database Security**:
   - Strengthen database security measures to protect sensitive personal information. This includes implementing encryption protocols and ensuring compliance with data protection regulations.
4. **Integration with APIs**:
   - Develop an API that will allow seamless integration with third-party systems for **form submissions**, **e-signatures**, and **digital KYC**. This will enable the system to serve a broader range of use cases in the future.
5. **Handling Low-Quality Images**:
   - Work on improving the OCR accuracy, especially for images of lower quality or documents with complex layouts. Using advanced image preprocessing techniques will help extract text more reliably.
6. **Scalability and Performance Optimization**:
   - Focus on optimising the performance and scalability of the system to handle large volumes of data efficiently, which will be necessary as the system is deployed in real-world scenarios.

By addressing these areas in the future, we aim to enhance the capabilities and reliability of **VerbalEx**, making it a comprehensive solution for digital document management and automation.

6.3 Conclusion:-

In this report, we have presented **VerbalEx**, an automated solution for extracting user data from government-issued ID cards using Optical Character Recognition (OCR). By integrating advanced OCR algorithms, image preprocessing, and modular architecture, the system ensures high accuracy in text extraction, even from varied document types.

Key findings include:

- The system effectively handles varying image quality, noise, and complex backgrounds to provide accurate and structured text extraction.
- **VerbalEx** is designed to be scalable, with API integration that allows future use in form submissions, e-signatures, and digital KYC processes.

While the system performs well in most scenarios, challenges such as low-quality images and complex layouts remain. Future improvements could focus on enhancing OCR accuracy in challenging conditions and adding machine learning models for error correction.

In summary, **VerbalEx** offers a powerful, scalable solution for automating ID data extraction, with potential applications in document digitization, data management, and accessibility. The integration of an API paves the way for its use in digital KYC, e-signatures, and form submissions in the future.

References:-

Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J. Wu, Andrew Y. Ng, "Text Recognition Using Convolutional Networks", *Computer Science Department Stanford University*, 353 Serra Mall, Stanford, CA 94305 USA: https://www.cs.utexas.edu/~dwu4/papers/TextRecog.pdf.

Axios Documentation : https://axios-http.com/docs/intro.

Express.js Documentation : https://expressjs.com/en/starter/installing.html.

International Journal of Computer Applications Technology and Research, Volume 3–Issue 4, 239 - 243, 2014, ISSN: 2319–8656 : https://ijcat.com/archives/volume3/issue4/ijcatr03041009.pdf.

MongoDB Documentation : https://docs.mongodb.com/manual/.

Mongoose Documentation : https://mongoosejs.com/docs/.

Node.js Documentation : https://nodejs.org/en/docs/.

React Documentation : https://legacy.reactjs.org/docs/getting-started.html.

React Router Documentation : https://reactrouter.com/web/guides/quick-start.