

DAILY DAIRY
Of
Summer Internship
B.Tech (CSE) - 5th Semester

Submitted by:

Kushagra

23124053

Internship At:

Global Holani Tradelink

1 month (Remote)



Department of Information Technology
Dr. B. R. Ambedkar National Institute of Technology
Jalandhar, Punjab - 1440011
(June-July 2025)

-> **June 9, 2025 (Monday):** Began the internship with an online kickoff meeting. Introduced to the mentor and team via Zoom, reviewed project scope and deliverables. Set up the development environment (installed Python, Git, required libraries) and configured access to the company's repository on GitHub. Completed onboarding paperwork and joined the company Slack channel.

-> **June 10, 2025 (Tuesday):** Collected detailed requirements by discussing use cases with the mentor. Researched existing document processing solutions. Started literature review on OCR technologies (Tesseract, Google Vision API, AWS Textract) and how they handle invoices and receipts. Prepared a brief report comparing options.

-> **June 11, 2025 (Wednesday):** Focused on hands-on experimentation: wrote and ran sample code using PyTesseract to extract text from scanned invoice images. Evaluated accuracy and speed on sample PDFs. Noted that Tesseract worked well on clear prints but struggled with noisy or rotated scans. Documented findings.

-> **June 12, 2025 (Thursday):** Investigated Large Language Models (LLMs) and prompt-based parsing. Signed up for an OpenAI developer account. Experimented with the OpenAI GPT-4 API using a Jupyter notebook: fed it a paragraph of invoice text to see if it could identify key fields. Reviewed LangChain documentation to learn how to chain prompts for OCR tasks.

-> **June 13, 2025 (Friday):** Held a planning session via Zoom to outline the MVP architecture. Created a basic design: an OCR module followed by an LLM parsing module, wrapped in a backend API, with a simple front-end for demonstration. Defined the first sprint tasks and set up the project board (Trello). Committed the initial project structure to GitHub and pushed code.

-> **June 16, 2025 (Monday):** Started Week 2 by designing the OCR pipeline in code. Implemented a Python script that takes an image file and returns its text using Tesseract. Modularized it into a function with configurable language and image preprocessing steps.

-> **June 17, 2025 (Tuesday):** Continued developing the OCR functionality. Added image preprocessing (grayscale conversion, thresholding) to improve recognition accuracy. Tested on various document samples (invoices, receipts, forms). Collected test results and tweaked parameters.

-> **June 18, 2025 (Wednesday):** Developed the backend service. Set up a Flask web server with an endpoint `/api/extract` that accepts file uploads. Integrated the OCR module so that uploaded images are processed and the extracted raw text is returned as a JSON response. Tested the API locally using Postman.

-> **June 19, 2025 (Thursday):** Refined the API by adding error handling (e.g. unsupported file types) and logging. Wrote documentation strings for the functions. Started writing unit tests for the OCR function to ensure consistent output. Committed changes and pushed to GitHub; created a pull request for review.

-> **June 20, 2025 (Friday):** Completed Week 2 with a working OCR endpoint. Spent the day reviewing and merging code. Attended a sprint demo meeting to show the current OCR capability. Received feedback to next focus on integrating the AI parsing stage. Planned the LangChain/LLM integration for the following week.

-> **June 23, 2025 (Monday):** Initiated Week 3 by setting up LangChain in the project. Installed the LangChain library and created a simple chain: a prompt template that instructs the model to parse invoice text into structured fields. Tested the LangChain + OpenAI chain on a sample invoice paragraph to confirm it outputs JSON-like key-value pairs.

- > **June 24, 2025 (Tuesday):** Worked on prompt engineering. Designed a system prompt that explains the document context ("This is an invoice from a textile supplier, extract Invoice No., Date, Total, and Item lines."). Tested different variations to improve reliability. Documented example prompts and outputs.
- > **June 25, 2025 (Wednesday):** Integrated the OCR and LLM stages in the API. Modified the /api/extract endpoint: after OCR extracts text, the text is passed to the LangChain chain, and the chain's output is returned. Verified end-to-end functionality with a sample image: the server now returns structured data (extracted fields).
- > **June 26, 2025 (Thursday):** Continued refining the AI parsing. Added post-processing to clean up the LLM output (e.g. strip quotes, format dates). Tested on multiple sample documents to identify any inconsistencies. Fine-tuned the prompt further for better accuracy on edge cases (handwritten digits, unusual layouts).
- > **June 27, 2025 (Friday):** Completed backend development phase. Finalized all API endpoints, including one for health check. Created a Python script to batch-test all sample documents in the test_docs folder and compare outputs. Fixed bugs discovered in this testing (such as missing fields). Prepared a mid-project report summarizing the back-end progress.
- > **June 30, 2025 (Monday):** Began Week 4 by planning the front-end. Chose React for the UI. Set up a new React project and designed wireframes: a simple page with a file upload button and a results table. Also discussed future features like multi-language support.
- > **July 1, 2025 (Tuesday):** Implemented the core front-end interface. Created a React component with a file input control. Wrote code to send the selected file to the backend /api/extract endpoint and receive JSON output. Added logic to display the results in a formatted way (table or list).
- > **July 2, 2025 (Wednesday):** Enhanced the UI for better user experience. Added progress indicators/spinner while processing. Improved form validation (e.g. only allow image/PDF uploads). Styled the page with CSS to make it more readable. Tested the full flow: upload a document in the browser, hit the API, and display the extracted fields.
- > **July 3, 2025 (Thursday):** Continued front-end development. Added a "Clear" button to reset the form. Enabled downloading the results as JSON. Handled error messages from the server (e.g. invalid file) and displayed alerts. Performed cross-browser testing to ensure compatibility.
- > **July 4, 2025 (Friday):** Mid-internship review with mentor. Demonstrated the working front-end and backend together. Received feedback: improve output formatting and test with more real-world samples. Began refining the styling and layout. Updated the project documentation to include the UI usage instructions.
- > **July 7, 2025 (Monday):** Entered final week of work. Focused on **testing and debugging**. Collected a new set of documents (invoices, receipts, business cards) and ran them through the system. Noted any failures: for example, text inverted in some PDFs. Fixed issues by rotating images and preprocessing. Began performance profiling (OCR time vs document size).
- > **July 8, 2025 (Tuesday):** Continued bug fixes and optimization. Optimized image preprocessing for faster OCR (resizing images). Improved prompt template clarity to handle multi-page documents. Added logging for tracking processing times and errors. Updated the GitHub README with installation and usage steps.

-> **July 9, 2025 (Wednesday):** Final day: wrapped up all loose ends. Conducted a final demo for the team, showing the platform working end-to-end with one-click extraction. Prepared and submitted the final internship report (this document). Committed the final code changes and pushed all branches. Received feedback and thanked the team for the opportunity.

Dashboard [Upload Document] [Export] [History] [Templates] [AI Assistant] [Logout]

Total Documents
0
All uploaded documents

Approved Documents
0
Documents ready for export

Total Exports
0
Number of exports performed

Recent Activity
No recent activity.

Export Activity
Latest document uploads and approvals
No documents yet

AI Assistant
Hello! I can help you with document processing, data extraction, and export management. What would you like to know?

Dashboard [Upload Document] [Export] [History] [Templates] [AI Assistant] [Logout]

Template Management
Create and manage your form templates

Available Templates [Create New Template]

Default Templates

Register Entry - Basic [Edit] [Use Template]
Basic register entry with Supplier, Party, Bill Number, Date, and Amount
Fields included: Supplier, Party, Bill Number, Register Date, Amount

Register Entry - With GST [Edit] [Use Template]
Register entry with additional GST Percentage field
Fields included: Supplier, Party, Bill Number, Register Date, Amount, Gst Percentage

AI Assistant
Get help with document processing and management

AI Assistant
Hello! I can help you with document processing, data extraction, and export management. What would you like to know?

Ask me anything... [Send]