FINAL INTERNSHIP REVIEW

Adish Kerkar – NITK

Srivatsa Kuruvadi – VIT AP

Sudharshan Srinivas M – VIT Vellore

OCR Engine

- The OCR Engine project aims to develop a web application to provide Optical Character Recognition (OCR) capabilities for extracting data from PDF documents.
- It allows users to handle PDF documents by extracting specific information based on user-defined keys.
- It also allows the user to convert tables from PDFs and turn them into Excel files that can be edited easily.

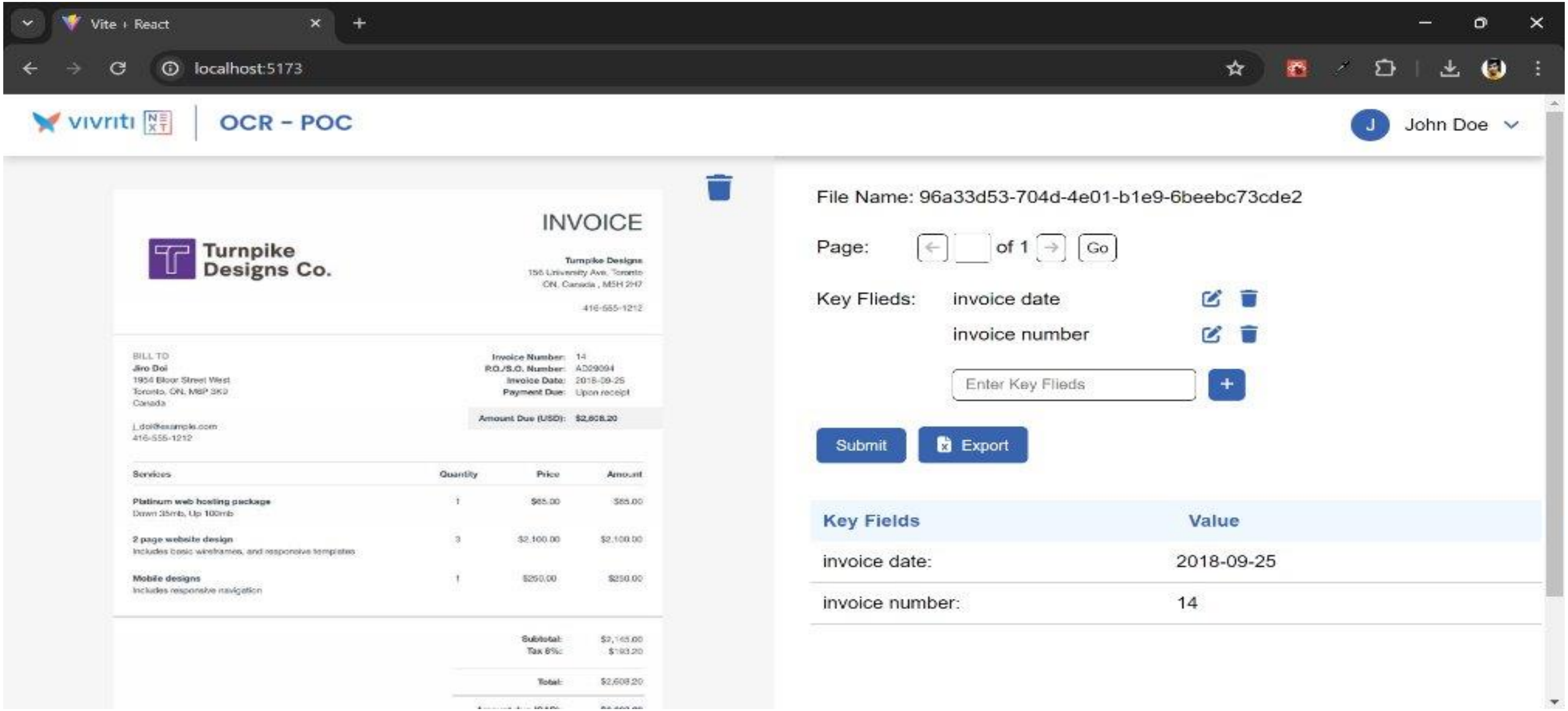
Technology stack utilized in the project:

React js- React is used in the project to create the frontend user interface where users can upload PDF documents and navigate through different pages. Users provide their input and the corresponding results provided by the tesseract is displayed.

Spring Boot- Spring Boot acts as the middleware that connects the React frontend with the backend Flask server. It handles tasks like rendering images for OCR and manages communication between the frontend and tesseract engine.

Flask- Flask serves as the backend framework hosting the Tesseract OCR engine. It processes image data from Spring Boot, performs OCR to extract text from PDFs or converts the table to excel and sends the results back to the React frontend for display.

User Interface- Generating Values for the given key



INVOICE

Turnpike Designs Co.

Turnpike Designs
156 University Ave, Toronto
ON, Canada , M5H 2H7

416-555-1212

BILL TO
Jiro Doi
1954 Bloor Street West
Toronto, ON, M6P 3K9
Canada

j_doi@example.com
416-555-1212

Invoice Number:	14
P.O./S.O. Number:	AD29094
Invoice Date:	2018-09-25
Payment Due:	Upon receipt
Amount Due (USD):	**$2,608.20**

Services	Quantity	Price	Amount
Platinum web hosting package Down 35mb, Up 100mb	1	$65.00	$65.00
2 page website design Includes basic wireframes, and responsive templates	3	$2,100.00	$2,100.00
Mobile designs Includes responsive navigation	1	$250.00	$250.00

Subtotal:	$2,165.00
Tax 8%:	$193.20
Total:	$2,608.20

File Name: 96a33d53-704d-4e01-b1e9-6beebc73cde2

Page: ← of 1 → Go

Key Flieds: invoice date

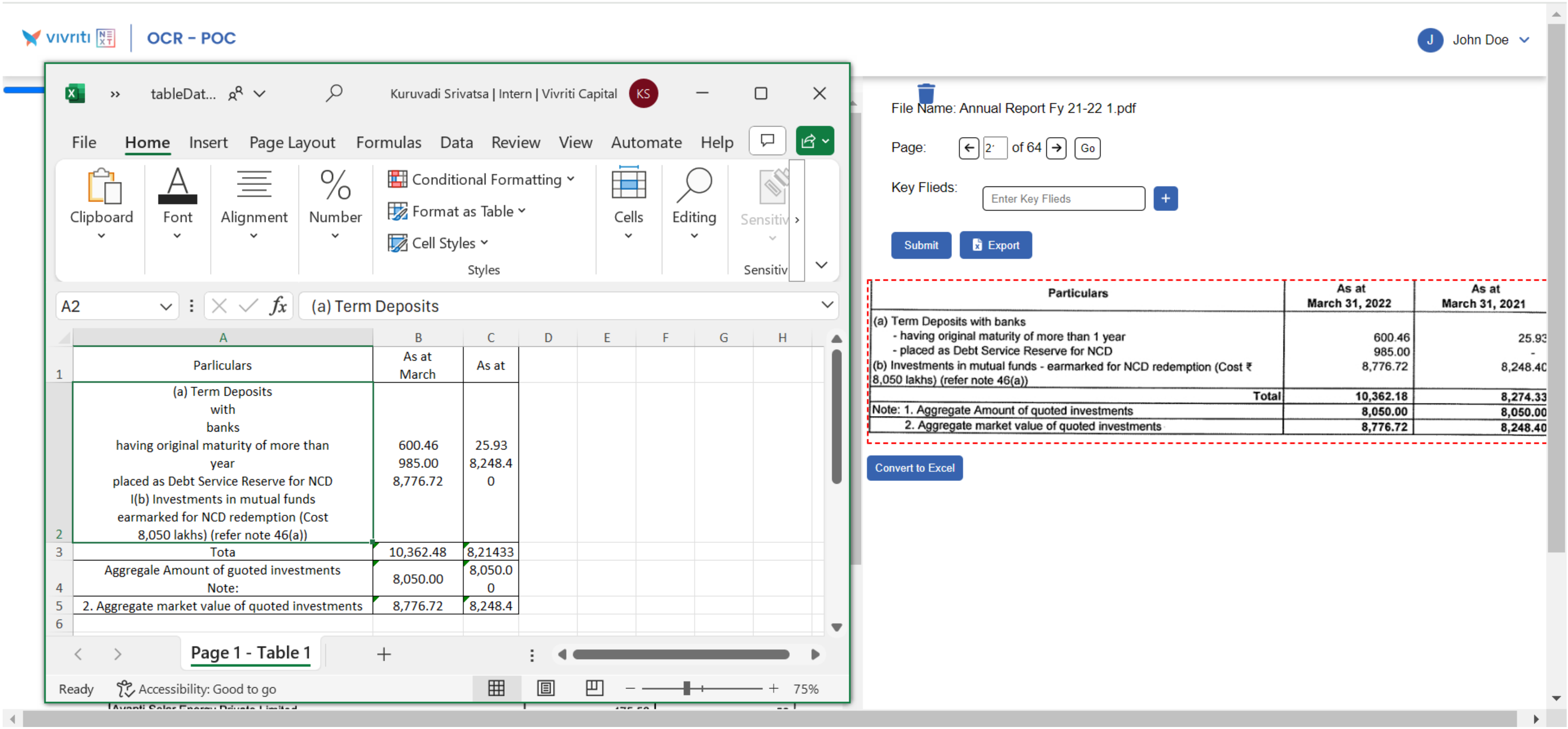
invoice number

Enter Key Flieds +

Submit Export

Key Fields	Value
invoice date:	2018-09-25
invoice number:	14

User Interface- Generating Excel file for selected table



A2 (a) Term Deposits

	A	B	C
1	Parliculars	As at March	As at
2	(a) Term Deposits with banks having original maturity of more than year placed as Debt Service Reserve for NCD I(b) Investments in mutual funds earmarked for NCD redemption (Cost 8,050 lakhs) (refer note 46(a))	600.46 985.00 8,776.72	25.93 8,248.4 0
3	Tota	10,362.48	8,21433
4	Aggregale Amount of guoted investments Note:	8,050.00	8,050.0 0
5	2. Aggregate market value of quoted investments	8,776.72	8,248.4
6			

File Name: Annual Report Fy 21-22 1.pdf

Page: 2 of 64

Key Flieds: Enter Key Flieds

Particulars	As at March 31, 2022	As at March 31, 2021
(a) Term Deposits with banks - having original maturity of more than 1 year - placed as Debt Service Reserve for NCD (b) Investments in mutual funds - earmarked for NCD redemption (Cost ₹ 8,050 lakhs) (refer note 46(a))	 600.46 985.00 8,776.72	 25.93 - 8,248.40
Total	10,362.18	8,274.33
Note: 1. Aggregate Amount of quoted investments	8,050.00	8,050.00
2. Aggregate market value of quoted investments	8,776.72	8,248.40

TIMELINE

WEEK 1

Tried to understand the functionality of nanonets website and tried to find libraries to render pdf in react

WEEK 2

Tried to use tesseract.js ocr to extract text on frontend but it didn’t give good results

WEEK 3

Used pytesseract on backend and build the first feature.

WEEK 4

Converted the figma design to react ui .

WEEK 5

Added the second feature functionality.

WEEK 6

Demonstrated the project and added toggle button for better extraction of tables

WEEK 7

Re-deployed the new codes in ec2 instance

WEEK 8

Upgraded tesseract for accurate extractions

Contributions made during the Internship

Project: OCR-POC

- Developed a responsive, user-friendly interface by converting Figma designs into a functional React frontend. Researched and integrated libraries to render PDFs in React, implementing pagination for easy navigation.
- Researched OCR technologies to enhance text extraction accuracy and contributed to deploying the application on an EC2 instance
- **Project: Digital Signatures with PyHanko:**
- Developed a Python script using the PyHanko library to create digital signatures on PDFs, enhancing the security and authenticity of digital documents.
- Designed and implemented a Flask API to extract specific data such as PAN number, assessment year, and gross salary from PDFs using OCR techniques.

Project: Accreditation-credit

- Created a demo application for storing data in JSON files instead of a traditional database and integrated the backend with the frontend using Flask.

My learnings

- I’ve gained experience in frontend development with React.js and backend development with Flask, including API creation and integration
- I acquired knowledge and hands-on experience in deploying applications on Amazon EC2, understanding setup, configuration, and ensuring seamless cloud operation.
- I developed skills in OCR technologies for text extraction and creating digital signatures on PDFs.
- Contributing to live projects with large-scale code bases with AWS Code commit.

Contributions made during the Internship

Project: OCR-POC

- Developed the Spring Boot backend responsible for preprocessing images received from the React frontend, ensuring they are formatted appropriately for the OCR engine's needs.
- Implemented Python code for image preprocessing, optimizing it to increase OCR accuracy.
- Conducted research on OCR technologies and explored different tools. Contributed to deploying the application on an EC2 instance.

Project: Credit-profile360-api

- Integrated a 'Convert to Excel' feature in Credit-Profile360 API's Spring Boot backend, to enable export of financial data from the User Interface

Project: Accreditation-credit

- Developed a demo environment for the "accreditation-credit-ui" project, including a Flask API backend with RESTful endpoints for efficient management of HTTP requests and responses.

My learnings

- I've learned how to use AWS Code Commit for managing code repositories on AWS and collaborative development practices.
- I gained hands-on experience in deploying applications on EC2 instances, understanding the setup and configuration required to host applications in the cloud.
- I acquired knowledge in implementing Tesseract for OCR tasks, enabling me to extract text from images and integrate this capability into applications for automated text recognition.
- Contributing to live projects with large-scale code bases.

Contributions made during the Internship

Project: OCR-POC

- Developed the Flask API responsible for collecting images and keys from the Spring backend, perform OCR and send back the extracted data to backend.
- Implemented Python codes for text extraction, regular expression matching and converting images to tables.
- Deployed the codes in EC2 instance with corresponding dependencies.

Project: EDMS Read

- Implemented Python code to extracts file and folder information from the EDMS and save it to an Excel file for faster verification of files.

Project: E-KYC

- Trained a Robust deep learning model which can accurately classify the deep fake videos and real videos using the PPG signals extracted from face.
- Integrated the model with a live streaming platform to conduct E-KYC and verify the applicant.

My learnings

- I’ve learnt developing RESTful APIs using Flask and implementing OCR techniques for text extraction from images.
- Gained hands-on experience in deploying applications on Amazon EC2, managing dependencies and configurations to ensure smooth operation on cloud platforms.
- Improved ability to automate file system tasks
- Gained experience in training and fine-tuning deep learning models using physiological signals.
- Improved adaptability by working on diverse projects, from OCR and data extraction to deep learning and real-time video processing.

Project 2- E-KYC

- The E-KYC project aims to create a web application that enables real-time KYC verification by conducting live meetings and recording customer videos during the session.
- It utilizes a deep learning model to analyze recorded videos and determine their authenticity by detecting deepfakes.
- It Automatically processes and sends real-time feedback on video authenticity within the platform's chat section.

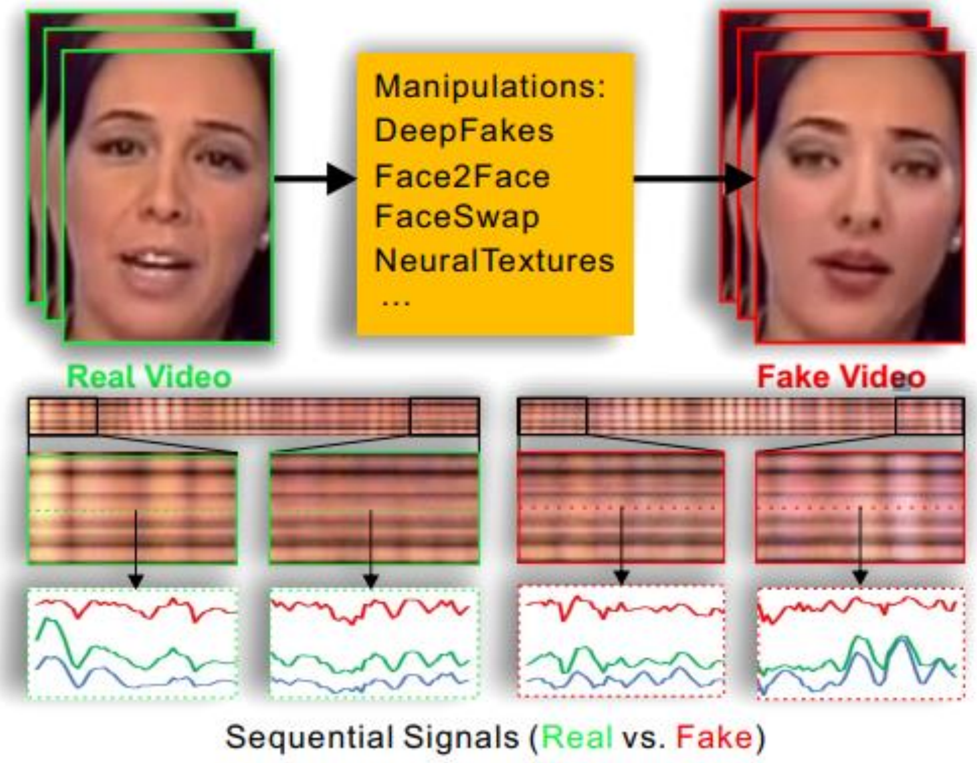
Technology stack utilized in the project:

React js- React is used in the project to create the frontend user interface where users can create and join meetings for real-time KYC, chat with the customer.

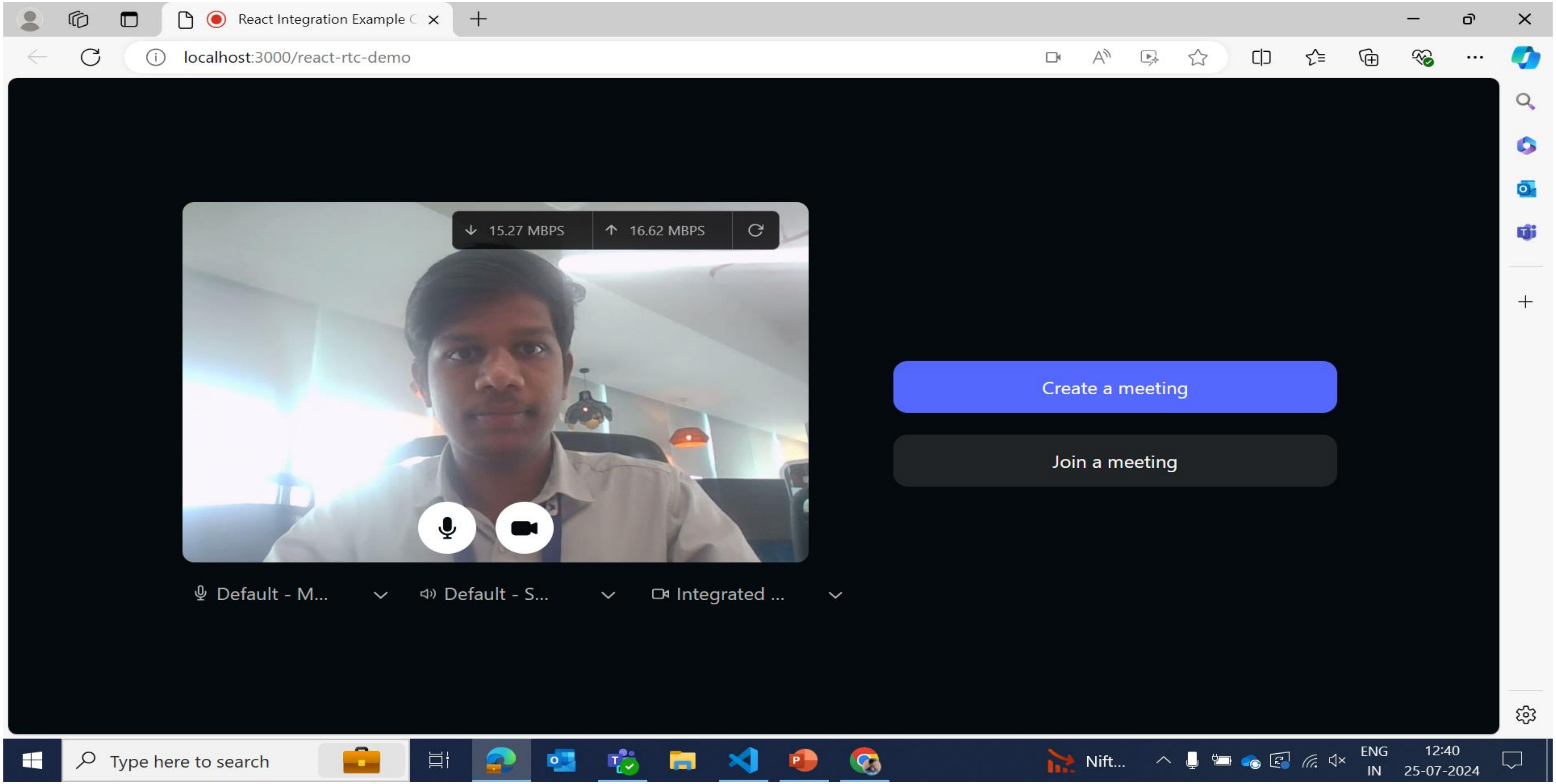
Python- Python serves as the backend framework hosting the deep learning model that is used to predict the authenticity of the customer, it collects the video and preprocesses it and sends the video to model for prediction.

Idea used for catching deepfakes

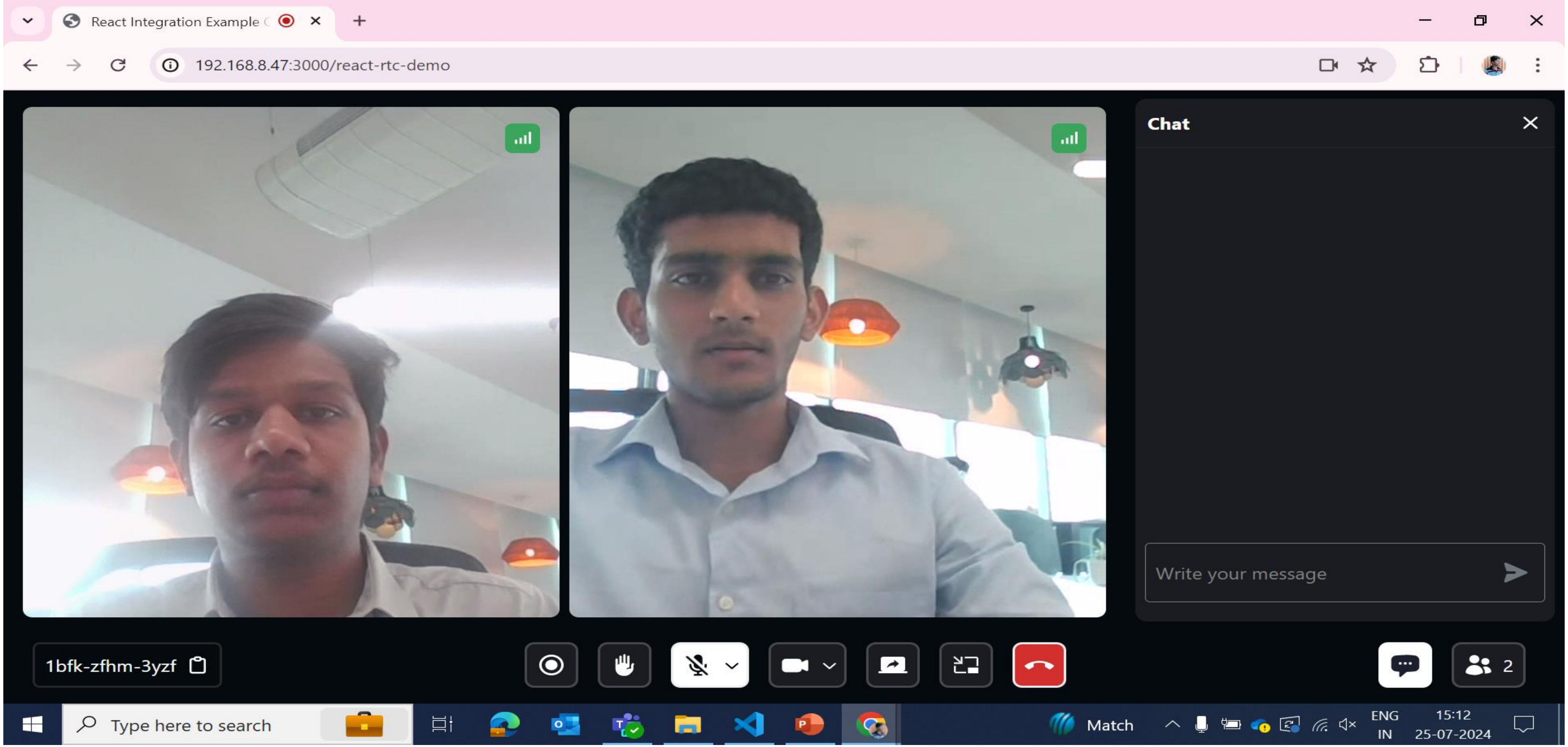
- We look for authentic clues in real videos, by assessing what makes us human— subtle blood flow in the pixels of a video.
- When our hearts pump blood, our veins change color. These blood flow signals are collected from all over the face and algorithms translate these signals into spatiotemporal maps.
- Then, using deep learning, we can instantly detect whether a video is real or fake.



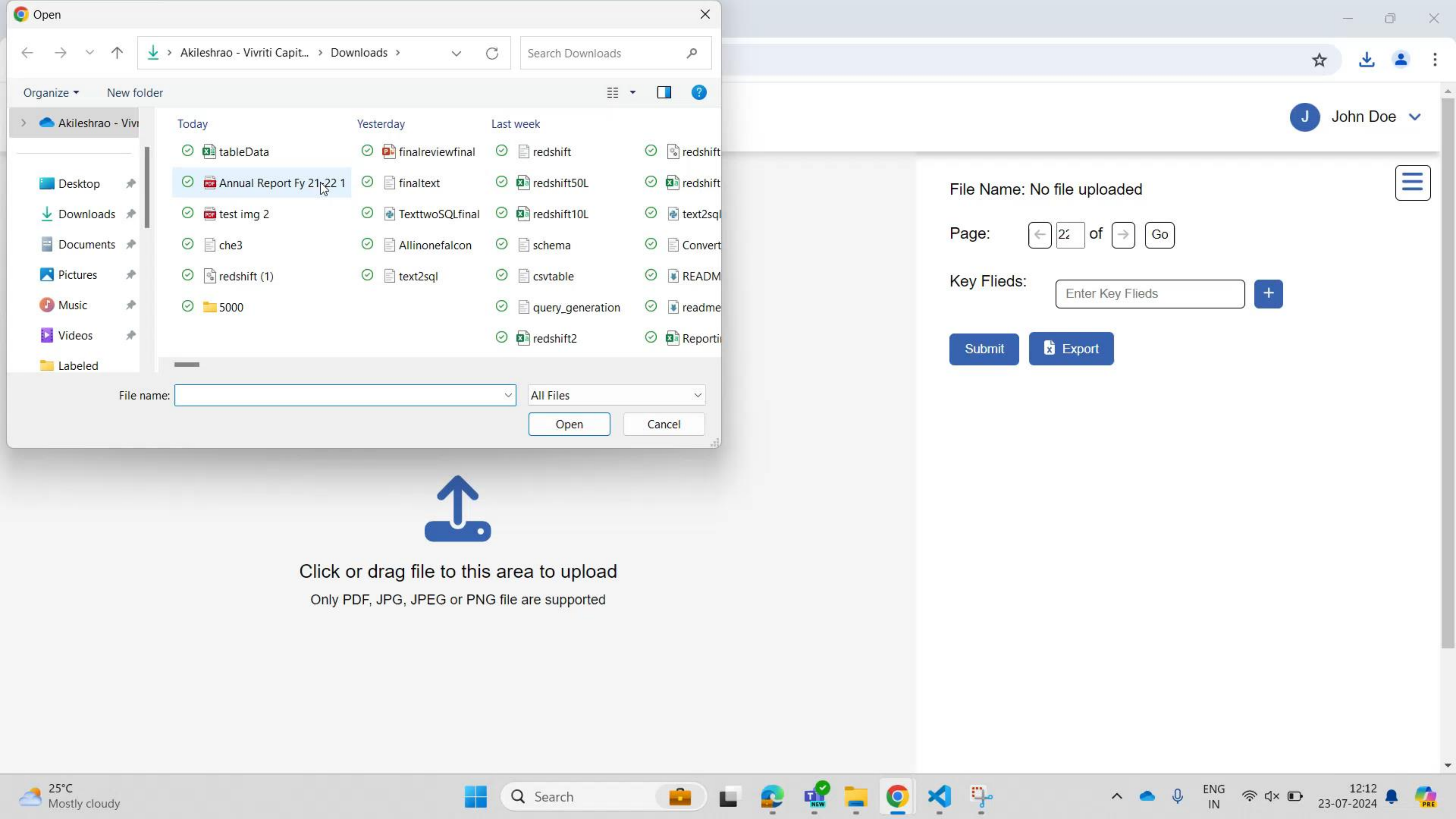
User interface – Main screen



User interface – Meeting screen



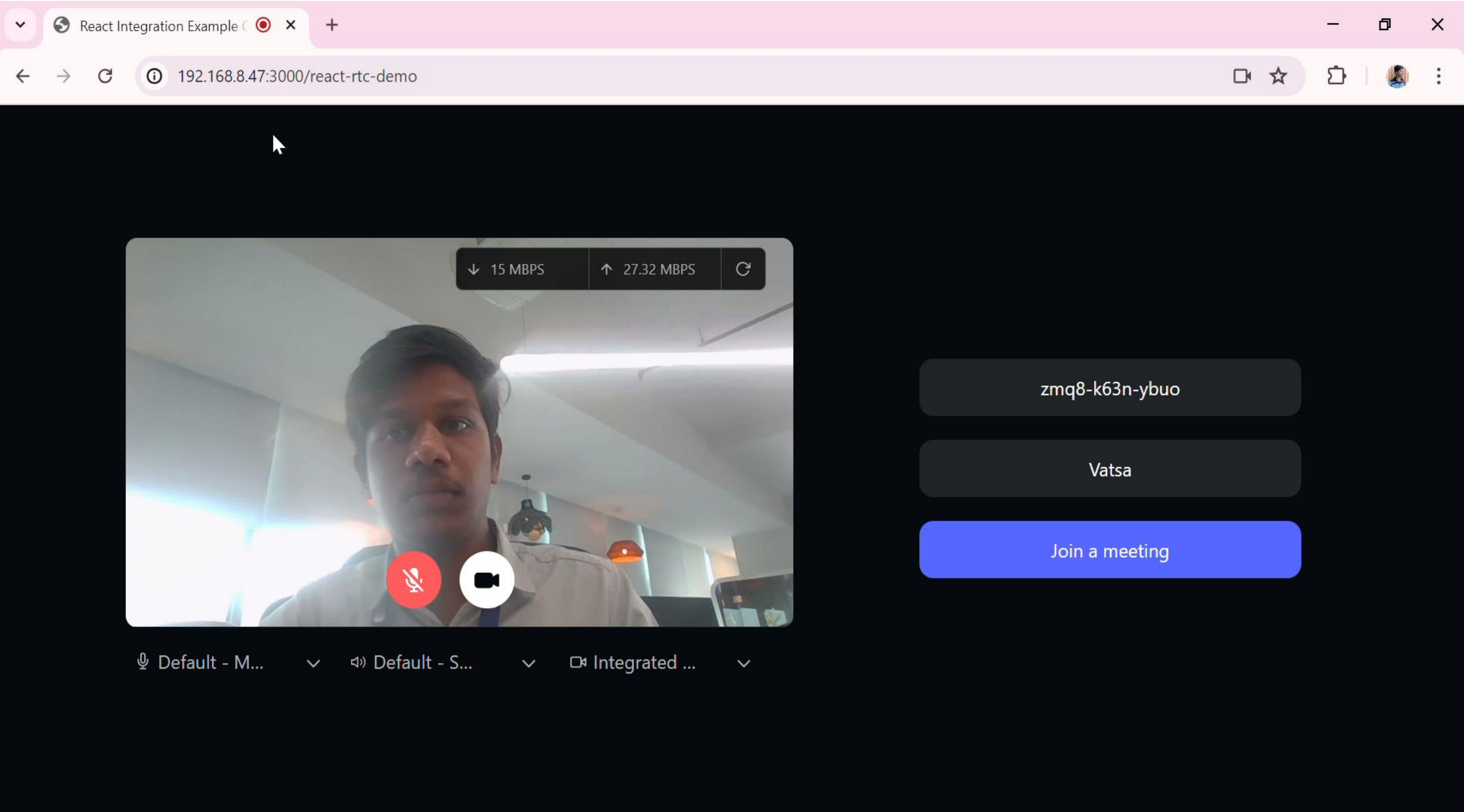
Thank you



File Name: No file uploaded

Click or drag file to this area to upload

Only PDF, JPG, JPEG or PNG file are supported



React Integration Example (

192.168.8.47:3000/react-rtc-demo

15 MBPS

27.32 MBPS

zmq8-k63n-ybuo

Vatsa

Default - M...

Default - S...

Integrated ...