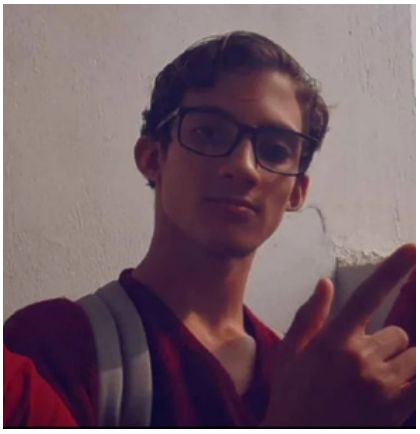

FARES AHMED

front end developer

A frontend web developer skilled in HTML, CSS, SCSS, JavaScript, Bootstrap, Git, GitHub, Axios, and Postman. Currently learning React and experienced in building full-stack projects with a focus on functionality and clean code

CONTACT

01013353405

fareselsisi35@gmail.com.

https://www.linkedin.com/in/fares-elsisi-613732321/

https://faresahmed.netlify.app/

EDUCATION

El Zagazig University (2024-2025)

Ain Shams University(2025-2026)

Bachelor of Education – Faculty of Specific Education
(2024-2028)

SKILLS

react.js

javascript

python

CSS

HTML

SCSS

bootstrap

Axois

postman

VS code

chrome dev tools

Git&Github

PROJECTS (TOP PROJECTS)

repos: https://github.com/Fares-Elsisi-2005

Social Media Lite (frontend)

LINK: https://chatbook123.netlify.app/

This is a simple social media web app built using HTML, CSS, and JavaScript.
It allows users to register, log in, and create, edit, or delete posts with images.
Data is managed using Axios to communicate with a REST API from the Backend.
The app supports user authentication, dynamic post display, and infinite scrolling.

Manhwa downloader (fullstack)

Status: Under development

A web scraping-based tool for seamless content access.

The frontend was built using HTML, CSS, and JavaScript, and it includes a form where the user enters the manhwa name and episode number. When the "Download" button is clicked, a fetch POST request is sent to the backend, and a progress bar is shown to indicate the current status. The backend, created with Node.js and Express.js, uses Puppeteer to automate the browser, search for the manhwa on the Webtoons website, and download the images. These images are then converted to base64 format and sent back to the frontend. Finally, the frontend uses the jsPDF library to compile the images into a single downloadable PDF file for the user

Manhwa Translator (fullstack)

Status: Under development

An OCR-powered application for translating content

Goal: a web application that enhances the manga reading experience by extracting text from images or PDFs, translating it from English to Arabic, and providing an interactive, user-friendly interface for both online and offline use.

Steps and Features

1. Text Extraction: Uses Tesseract.js to perform OCR, extracting text from manga images or PDFs with high accuracy.
2. PDF Processing: Employs PDF.js to process PDF files, rendering pages as images for text extraction.
3. Backend Processing: Built with Express.js and Node.js, the backend handles PDF processing, converting pages to base64 images and extracting text server-side.
4. Translation: Integrates an external API to translate extracted English text into Arabic, with fallback translations for offline use.
5. Caching: Stores translations and page data in IndexedDB for fast access and efficient offline functionality.
6. Interactive UI: Displays translated words in clickable boxes over images, showing original and translated text on click.
7. Offline Support: Generates HTML and JSON files for offline reading, allowing users to access translated manga without internet.
8. Image Optimization: Splits large images into smaller segments to ensure smooth performance and compatibility.