UNICONVERTER

Technology, Flow, and Format Support

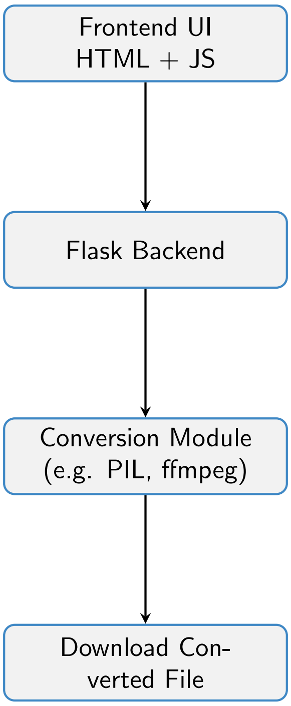
Badger Code

July 9, 2025

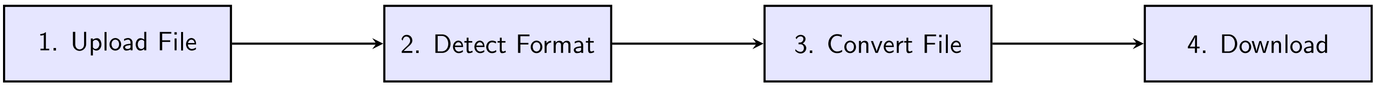
Universal File Converter

Universal File Converter is a web application designed to convert various file formats seamlessly. It supports a wide range of input and output formats, making it versatile for different user needs. The application features a user-friendly interface built with HTML and JavaScript, backed by a Flask server that handles file uploads, format detection, and conversion processes.

Technology Stack Flowchart



Application Flow Diagram



Supported Formats

File Type	Extensions
Image	jpg;jpeg;png;webp;gif;bmp;ico;svg;eps;tga;tif;tiff;ppm;xbm;icns
Audio	mp3;wav;flac;aac;m4a;ogg;opus;wma;aiff;alac;amr;mka
Video	mp4;mov;avi;webm;mkv;flv;wmv;mpeg;mpg;3gp;m4v;ts
Document	pdf;txt

Table 1: Supported File Types and Extensions

Code Walkthrough

0.1 Conversion Logic

```
@app.route("/convert", methods=["POST"])
def convert():
        """
        Convert uploaded files to the target format.
        """
        files = request.files.getlist("files")
        target_format = request.form.get("target_format")
        if not files or not target_format:
                return jsonify({"error": "No files or target format specified"}), 400

        # Convert each file
        converted_files = []
        for file in files:
                if file:
                        filename = secure_filename(file.filename)
                        path = os.path.join(UPLOAD_FOLDER, filename)
                        file.save(path)
                        file_info = {"filename": filename, "target_format": target_format}
                        result = convert_one(file_info)
                        converted_files.append(result)

        # Zip converted files if more than one
        if len(converted_files) > 1:
                zip_filename = f"converted_{uuid.uuid4().hex}.zip"
                zip_path = os.path.join(CONVERTED_FOLDER, zip_filename)
                with zipfile.ZipFile(zip_path, 'w') as zipf:
                        for converted_file in converted_files:
                                file_path = os.path.join(CONVERTED_FOLDER, converted_file)
                                zipf.write(file_path, arcname=converted_file)
                return send_file(zip_path, as_attachment=True)
        else:
                return send_file(os.path.join(CONVERTED_FOLDER, converted_files[0]), as_attachment=T
```