DIGVIJAY NATH POST GRADUATE COLLEGE GORAKHPUR

SESSION 2024-25

CIVIL LINES, GORAKHPUR(B++ With C.G.P.A 2.84 accredited with NAAC)

This is to certify that project work entitled

“FILESWITCH”

Has been carried out successfully

HOD **Submitted To**

(Dr. Harishankar Gupta) **(Dr. Pawan kumar Pandey)**

Submitted by

1.Student Name : Abhay Pandey(2514057260059)

2.Student Name: Aman Paswan(2514057260039)

Affiliated by D.D.U Gorakhpur University

AN ISO 9001:2015 Certified Institute

BALAJI ACADEMY

ISO 9001:2015

Reg. By: CR Act. Ministry of HRD Govt. of India

A National Program of Information Technology Education & Development

Regd. 780/2019-2020 Society Act Govt. of Uttar Pradesh

CERTIFICATE

This is to certify that

ABHAY PANDEY S/O VIRENDRA PANDEY

Has successfully completed the course specified and is awarded this certificate of achievement under the seal of institution.

Registration No. : R20240813438 Enrolment No. : E20244414050

Course Name : Developing Web Applications Using PHP

Period of Course : 3 Months

Grade Awarded : A

Issue Date : 14/12/2024

Scan & Verification

GRADE : A **RESULT : PASS**

Centre Head

Director Sign.

Add.: 3rd Floor, Vishwanath Tower, Medical College Road, Gorakhpur City Office: 10 Park Road, City Mall Civil Line, Gorakhpur

Visit Us: www.balajiacademygkp.com

Acknowledgement

I would like to express my sincere gratitude to all those who supported and guided me throughout the completion of this project.

First and foremost, I am deeply thankful to Pawan Kumar Pandey Sir, whose constant encouragement, expert guidance, and insightful feedback were invaluable throughout this journey. Their mentorship played a crucial role in shaping the direction and outcome of this work.

I would also like to extend my heartfelt thanks to my peers and friends who provided useful suggestions and moral support during various stages of the project. Their collaboration and shared insights contributed meaningfully to the success of this endeavor.

Additionally, I express my gratitude to my family for their unwavering support, patience, and understanding, which helped me stay focused and motivated.

Lastly, I acknowledge the use of resources, tools, and technologies that made this project possible. Special thanks to [mention any APIs, frameworks, libraries, etc., like Firebase, Node.js, ConvertAPI, etc.], which were instrumental in the development process.

This project has been a great learning experience, and I am grateful to everyone who played a part in it.

Abhay Pandey

Aman Paswan

Objective of the Project

The primary objective of this project is to develop a **full-stack web application** that enables users to **upload files and convert them between various formats** such as **DOCX, PDF, JPG, and PNG** in a seamless, efficient, and user-friendly manner.

This project aims to:

1. **Provide an intuitive user interface** for uploading, previewing, converting, and downloading files using modern web technologies.
2. **Ensure compatibility and responsiveness** across different devices and screen sizes using HTML, CSS, and React.
3. **Implement a scalable backend** using Node.js and Express to handle file processing, format conversion logic, and API integration with third-party services (e.g., ConvertAPI, CloudConvert).
4. **Incorporate security and validation mechanisms** to protect file uploads, prevent abuse, and ensure reliable operations.
5. **Demonstrate integration of both modern (Node.js) and traditional (PHP) backend technologies**, offering flexibility in development and deployment.
6. **Offer real-time feedback and seamless communication** between frontend and backend through asynchronous JavaScript and RESTful APIs.
7. **Deliver a complete and extensible solution** that can be enhanced to support additional file formats, authentication features, or cloud storage in the future.

Scope of the Project

This project provides a web-based file conversion tool that allows users to upload files in one format and convert them into other supported formats. It is designed with modularity and scalability in mind, combining a modern frontend with a robust backend system.

✅ Functional Scope:

1. **File Upload Interface**
    - Users can upload documents (DOCX, PDF) and images (JPG, PNG).
    - Real-time previews are available for supported formats.
2. **Multi-Format Conversion**
    - Converts files between DOCX ↔ PDF, PDF ↔ JPG/PNG, and image formats ↔ DOCX/PDF using third-party APIs.
3. **Downloadable Output**
    - After conversion, users can download the result directly.
4. **Frontend Technologies**
    - Built with React for a responsive, component-based UI.
    - Styled with CSS for a clean and responsive experience.
5. **Backend Technologies**
    - Node.js and Express handle API requests and file processing.
    - PHP is optionally used for lightweight scripts or legacy logic.
6. **Extensibility**
    - Can be expanded to support user accounts, history tracking, cloud storage (e.g., AWS, Firebase), or additional file formats.

Limitations of the Project

While the project meets its primary objectives, there are a few limitations due to technical and time constraints:

⚠ Technical Limitations:

1. **Limited Format Support**
    - Only supports conversions between DOCX, PDF, JPG, and PNG. Other formats like XLSX, PPTX, or TXT are not currently supported.
2. **File Size Restrictions**
    - Large files (e.g., over 25MB) may fail or take longer due to API or server limitations.
3. **Dependency on External APIs**
    - Relies on services like ConvertAPI or CloudConvert, which may have rate limits or require paid plans for high-volume usage.
4. **No User Authentication** *(optional limitation if not implemented)*
    - File conversions are anonymous; there is no user account, login, or history tracking feature yet.
5. **No Cloud Storage Integration**
    - Files are stored temporarily and not saved in cloud services like Google Drive or AWS S3.
6. **Minimal Error Handling**
    - Basic error messages are displayed, but detailed logging or user-friendly error explanations could be improved.

Hardware and Software Requirement

Hardware Required

Processor-Intel Core i5

Ram-16GB

SSD- 512GB

Software Required

Frontend-HTML,CSS,JAVASCRIPT,REACT

Tailwind

Backend- Express.js,Node.js,convertapi

Tech Stack Used In Project

1. HTML (HyperText Markup Language)

HTML is the foundation of all web pages. It provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.

Key Features:

- **Semantic Structure**: HTML5 introduces semantic tags like `<header>`, `<footer>`, `<article>`, and `<section>` to improve code readability and accessibility.
- **Media Embedding**: Easily integrate images, videos, audio, and iframes.
- **Form Handling**: Provides elements like `<input>`, `<textarea>`, and `<select>` for creating interactive forms.

Usage in Project:

- HTML was used to build the basic structure of the user interface including the layout for file uploads, buttons, and forms.
- Semantic tags were implemented to maintain accessibility and SEO best practices.

2. CSS (Cascading Style Sheets)

CSS is responsible for the visual styling of the HTML content. It controls layout, color, fonts, spacing, and overall presentation.

Key Features:

- **Selectors & Specificity**: Enables targeting of specific HTML elements for styling.
- **Responsive Design**: Media queries allow for adaptable designs on different screen sizes.
- **Flexbox & Grid**: Modern layout models that offer flexible and powerful ways to align and distribute space.
- **Custom Properties**: CSS variables to reuse values across stylesheets.

Usage in Project:

- CSS was used to make the user interface visually appealing and responsive.
- Flexbox and media queries were applied to ensure cross-device compatibility.
- CSS animations and transitions improved user experience and interactivity.

3. JavaScript

JavaScript is a scripting language that enables interactive web functionalities. It allows the creation of dynamic updates, control over multimedia, and interaction handling.

Key Features:

- **DOM Manipulation**: Enables dynamic content changes without reloading the page.
- **Event Handling**: Listens to and responds to user interactions such as clicks, hovers, and form submissions.
- **Asynchronous Programming**: Promises, async/await help handle API calls efficiently.
- **Modularization**: Modern JavaScript (ES6+) supports modules, classes, and arrow functions, enhancing code structure and reusability.

Usage in Project:

- Handled user interactions like uploading a file, clicking convert/download buttons, and displaying previews.
- Communicated with backend APIs asynchronously using `fetch` or `axios`.
- Utilized JavaScript modules to organize the code cleanly.

4. React.js

React is a JavaScript library developed by Facebook for building user interfaces, especially single-page applications where components dynamically update.

Key Features:

- **Component-Based Architecture**: UI is broken into reusable components.
- **Virtual DOM**: React uses a virtual DOM to minimize real DOM manipulations, improving performance.
- **Hooks**: Features like `useState`, `useEffect`, and custom hooks help manage state and side effects.
- **JSX Syntax**: HTML-like syntax in JavaScript, enabling clean and expressive UI code.

Usage in Project:

- Built a component-based frontend with components for file upload, format selection, preview, and result display.
- Used `axios` in `useEffect` for API interactions.
- Implemented conditional rendering and state management for smooth UI/UX transitions.

5. Node.js

Node.js is a JavaScript runtime built on Chrome's V8 engine that allows developers to run JavaScript on the server side.

Key Features:

- **Non-Blocking I/O**: Handles thousands of concurrent requests using asynchronous callbacks.
- **NPM (Node Package Manager)**: Comes with a large ecosystem of open-source packages.
- **Event-Driven Architecture**: Enables scalable and efficient applications.
- **Built-In Modules**: Offers core modules like `fs`, `http`, `path` for file and network handling.

Usage in Project:

- Served as the server-side runtime for handling API requests and responses.
- Used to create an Express server to route client requests.
- Managed file upload, format conversion logic, and temporary storage.

6. Express.js

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Key Features:

- **Routing**: Simplifies URL routing for handling different HTTP requests.
- **Middleware Support**: Process requests before reaching the endpoint.
- **Request/Response Handling**: Built-in methods to easily manage APIs.
- **Integration**: Works seamlessly with templating engines, databases, and third-party libraries.

Usage in Project:

- Created RESTful API endpoints for:
  - Uploading files
  - Converting files to selected formats
  - Sending back download links
- Used `multer` for handling file uploads.
- Applied middleware for CORS, logging, and error handling.

7. PHP (Hypertext Preprocessor)

PHP is a **server-side scripting language** traditionally used for building dynamic web pages and backend services. It can integrate directly with HTML and is particularly useful for server-side form handling, session management, and database interaction.

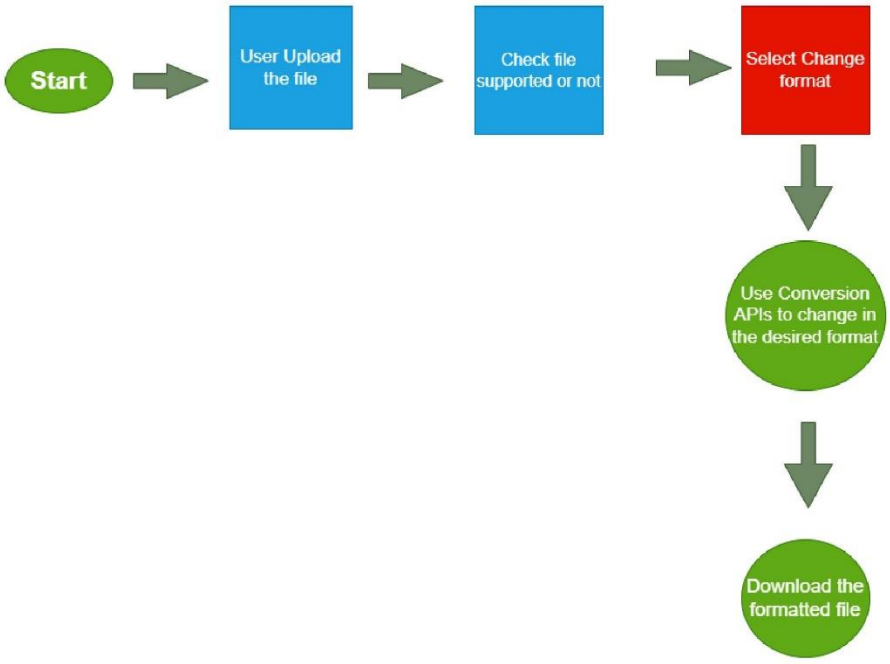
✅ Key Features:

- **Embedded in HTML**: Mixes PHP logic within HTML templates.
- **Server-Side Scripting**: Good for handling form submissions and session management.
- **Database Connectivity**: Seamlessly integrates with MySQL, PostgreSQL, etc.
- **Lightweight and Fast**: Quick to set up, especially for small APIs and utility scripts.

🔧 Use in Project (if applicable):

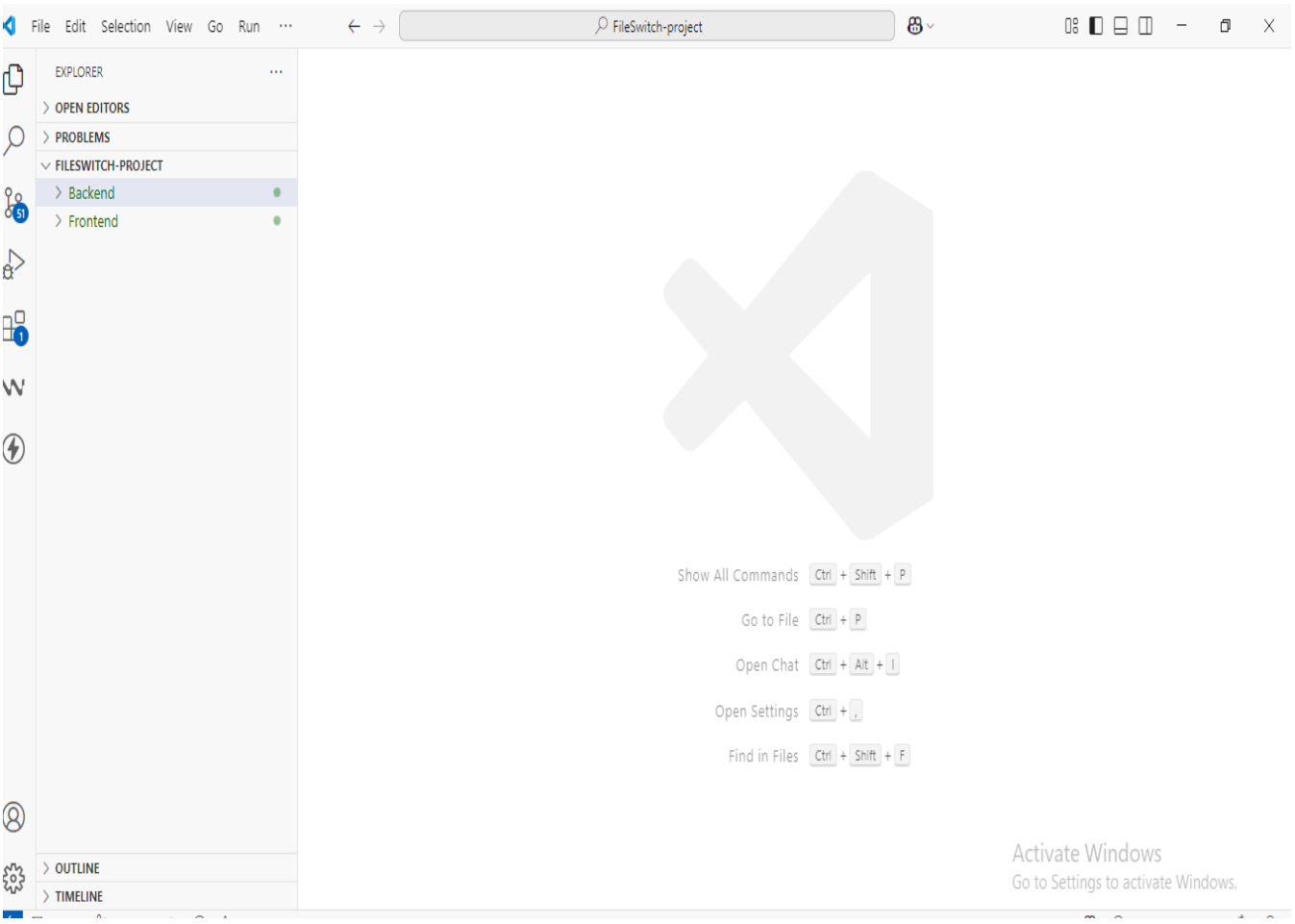
- PHP was used for:
  - A lightweight admin panel.
  - Server-side file validation.
  - Communicating with a MySQL database for logging user data or file metadata.
- It was helpful for writing quick test scripts or backend utilities that didn't require a full Node.js stack.

Process Description:

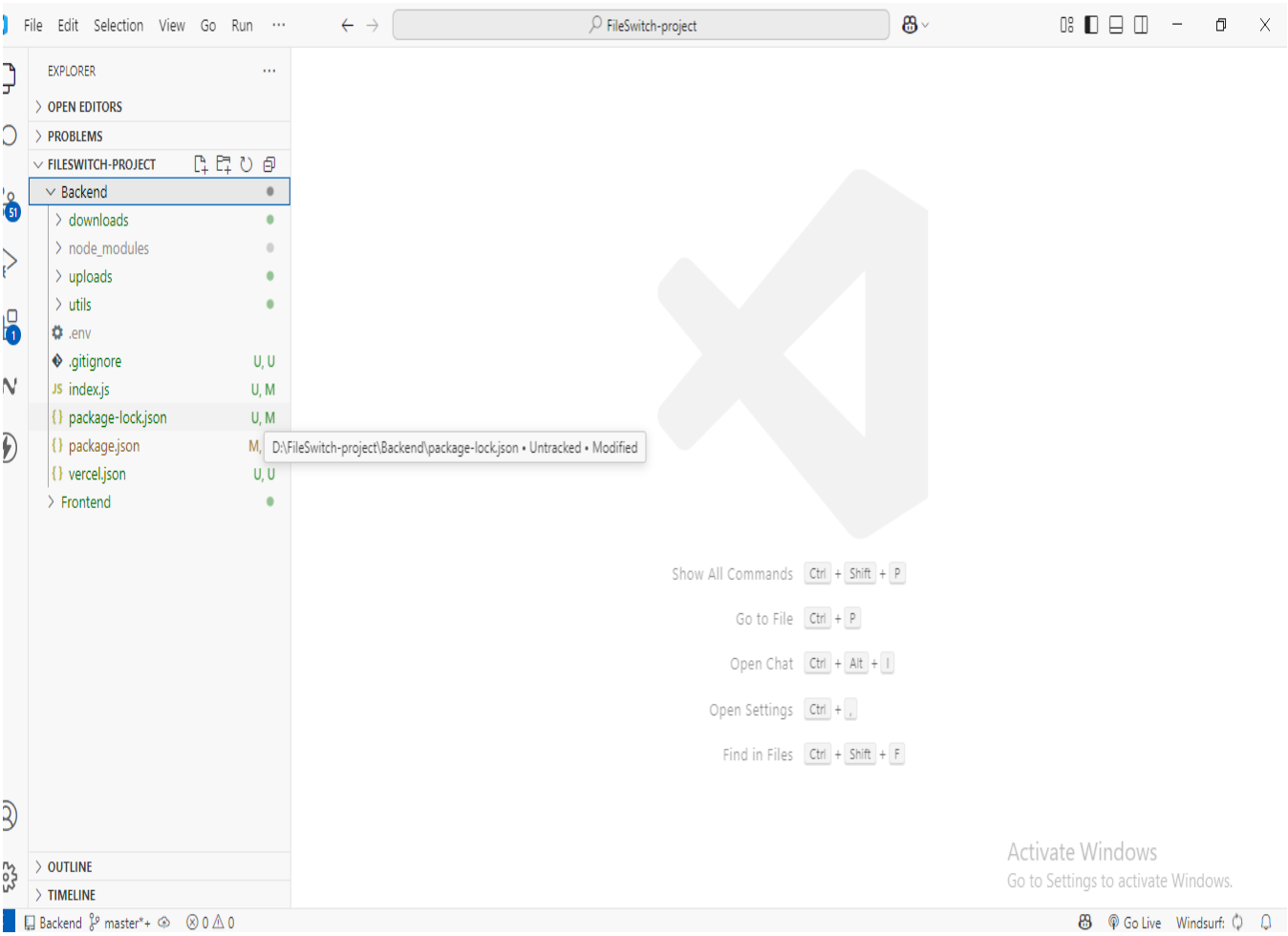


UML Diagram of Process

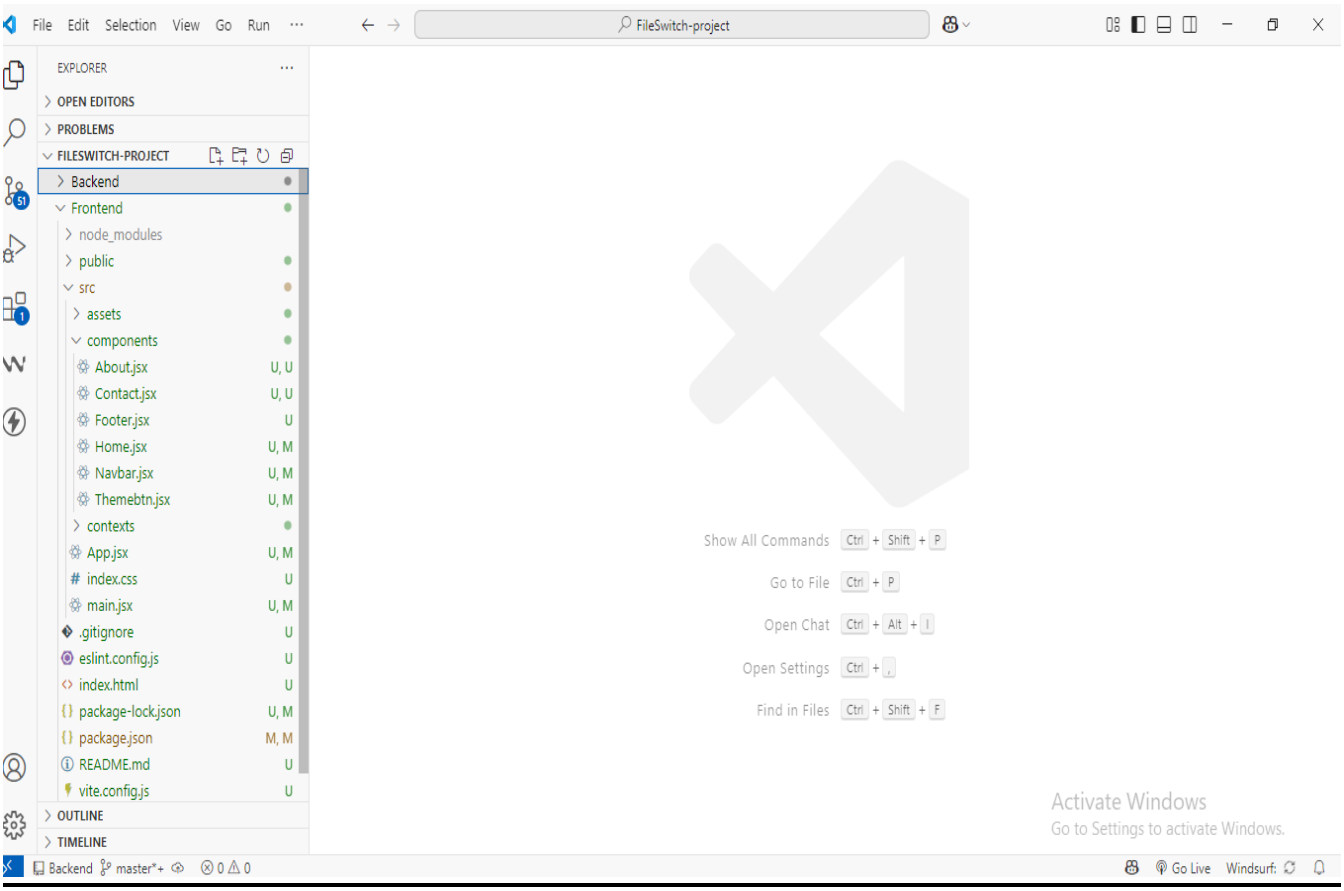
Folder Structure



Backend Structure



Frontend Structure



Frontend Source Code

Navbar Component Source Code

```
import { useEffect, useState } from "react";
import { ThemeProvider } from "../contexts/theme.js";
import ThemeBtn from "../components/Themebtn";
import { FaBars, FaTimes } from "react-icons/fa";



function Navbar({ setPage }) {
 const [themeMode, setThemeMode]=
useState("light");
  const [isMenuOpen, setIsMenuOpen] =useState(false);

  const lightTheme = () => setThemeMode("light");
  const darkTheme = () => setThemeMode("dark");

  useEffect(() => {
   document.querySelector("html").classList.remove("
light", "dark");
document.querySelector("html").classList.add(
themeMode);
  }, [themeMode]);

  return (
    <ThemeProvider value={{ themeMode, lightTheme,
darkTheme }}>
      <div className="fixed top-0 w-full bg-white dark:bg-
gray-900 text-black border-b-2 dark:text-white shadow-lg z-
50">
        <div className="max-w-screen-2xl mx-auto flex items-
center justify-between px-4 py-3 md:px-10">
          <h1 className="text-2xl font-bold">
            File<span className="text-3xl text-green-
500">To</span>File
          </h1>

          {/* Desktop Nav */}
```

```
          <div className="hidden md:flex gap-6 items-
center">
            <button onClick={() => setPage('home')}
className="text-xl font-bold cursor-pointer hover:scale-110
transition-transform">Home</button>
            <button onClick={() => setPage('contact')}
className="text-xl font-bold cursor-pointer hover:scale-110
transition-transform">Contact</button>
            <button onClick={() => setPage('about')}
className="text-xl font-bold cursor-pointer hover:scale-110
transition-transform">About</button>
            <ThemeBtn />
          </div>

          {/* Mobile Menu Toggle */}
          <div className="md:hidden text-2xl cursor-pointer"
onClick={() => setIsMenuOpen(!isMenuOpen)}>
            {isMenuOpen ? <FaTimes /> : <FaBars />}
          </div>
        </div>

        {/* Mobile Menu */}
        {isMenuOpen && (
          <div className="md:hidden flex flex-col items-
start gap-4 px-6 pb-4 text-black dark:text-white font-bold
bg-white dark:bg-gray-900">
            <button onClick={() =>
setPage('home')}>Home</button>
            <button onClick={() =>
setPage('contact')}>Contact</button>
            <button onClick={() =>
setPage('about')}>About</button>
            <ThemeBtn />
          </div>
        )}
      </div>
    </ThemeProvider>
  );
}
```

```
export default Navbar;
```

Navbar Component

Home Component Source Code

```
import React, { useState } from "react";

import {
  FaFileWord,
  FaFilePdf,
  FaFileImage,
  FaFileAlt,
  FaFileExcel,
  FaEye,
  FaDownload,
  FaRedo
} from "react-icons/fa";
import axios from "axios";

const fileIcon = {
  docx: <FaFileWord className="text-3xl mr-3 text-blue-600"
/>,
  pdf: <FaFilePdf className="text-3xl mr-3 text-red-600" />,
  jpg: <FaFileImage className="text-3xl mr-3 text-purple-
600" />,
  png: <FaFileImage className="text-3xl mr-3 text-green-600"
/>,
  txt: <FaFileAlt className="text-3xl mr-3 text-gray-600"
/>,
  xlsx: <FaFileExcel className="text-3xl mr-3 text-green-
700" />

};

function Home() {
  const [selectedFile, setSelectedFile] = useState(null);
```

```
  const [conversionFormat, setConversionFormat] =
useState("");
  const [message, setMessage] = useState("");
  const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);


  // Conversion state
  const [isConverted, setIsConverted] = useState(false);
  const [previewUrl, setPreviewUrl] = useState("");
  const [convertedBlob, setConvertedBlob] = useState(null);

  const handleFileChange = (e) => {
    if (e.target.files[0]) {
      setSelectedFile(e.target.files[0]);
      resetConversionState();
    }
  };

  const handleConvertFormat = (e) => {
    setConversionFormat(e.target.value);
    resetConversionState();
  };

  const resetConversionState = () => {
    setIsConverted(false);
    setPreviewUrl("");
    setConvertedBlob(null);
    setMessage("");
    setError("");
  };

  const defaultFileIcon = (file) => {
    if (!file) return null;
    const fileExtension =
file.name.split(".").pop().toLowerCase();
    return fileIcon[fileExtension] || <FaFileAlt
className="text-3xl mr-3" />;
  };
```

```
  const conversionMap = {
    docx: { pdf: "docx-to-pdf", jpg: "docx-to-jpg", png:
"docx-to-png" },
    pdf: { docx: "pdf-to-docx", jpg: "pdf-to-jpg", png:
"pdf-to-png" },
    jpg: { docx: "jpg-to-docx", pdf: "jpg-to-pdf", png:
"jpg-to-png" },
    png: { docx: "png-to-docx", pdf: "png-to-pdf", jpg:
"png-to-jpg" }
  };

  const handleConversion = async () => {
    if (!selectedFile || !conversionFormat) return;

    const inputFormat =
selectedFile.name.split(".").pop().toLowerCase();
    const conversionPath =
conversionMap[inputFormat]?.[conversionFormat];

    if (!conversionPath) {
      setError("Invalid file conversion selected.");
      return;
    }

    setLoading(true);
    setError("");

    try {
      const formData = new FormData();
      formData.append("file", selectedFile);

      const response = await axios.post(
        `/convertFile/${conversionPath}`,
        formData,
        { responseType: "blob" }
      );

      // Store the blob for later download
```

```
      setConvertedBlob(response.data);

      // Create a preview URL
      const previewUrl =
URL.createObjectURL(response.data);   //checking the
diffference two previewUrl one is state hook and the other
one is varable
      setPreviewUrl(previewUrl);

      setIsConverted(true);
      setMessage("File converted successfully! You can
preview or download it.");
    } catch (error) {
      console.error("Conversion Error:", error);
      setError(error.response?.data?.message || "Conversion
failed");
    } finally {
      setLoading(false);
    }
  };


  //debug the below code line-by-line
  const handleDownload = () => {
    if (!convertedBlob) return;

    const url = URL.createObjectURL(convertedBlob);
    const link = document.createElement("a");
    link.href = url;
    link.download = selectedFile.name.replace(/\.[^/.]+$/,
`.${conversionFormat}`) //debug this line
    document.body.appendChild(link);
    link.click();
    link.remove();
    URL.revokeObjectURL(url);

    setMessage("File downloaded successfully!");
  };
```

```
  const resetAll = () => {
    setSelectedFile(null);
    setConversionFormat("");
    resetConversionState();
  };

  // Determine what preview component to show based on
format
  const renderPreview = () => {
    if (!previewUrl) return null;

    switch (conversionFormat) {
      case 'pdf':
        return (
          <div className="mt-4 w-full">
            <h3 className="text-lg font-semibold mb-
2">Preview:</h3>
            <iframe
              src={previewUrl}
              className="w-full h-96 border border-gray-300
rounded"
              title="PDF Preview"
            />
          </div>
        );
      case 'jpg':
      case 'png':
        return (
          <div className="mt-4 w-full">
            <h3 className="text-lg font-semibold mb-
2">Preview:</h3>
            <img
              src={previewUrl}
              alt="Image Preview"
              className="max-w-full max-h-96 border border-
gray-300 rounded mx-auto"
            />
          </div>
        );
```

```
      case 'docx':
        return (
          <div className="mt-4 w-full">
            <h3 className="text-lg font-semibold mb-
2">Preview:</h3>
            <div className="bg-gray-100 p-4 border border-
gray-300 rounded">
              <p className="text-gray-500">DOCX preview not
available. Please download to view the document.</p>
            </div>
          </div>
        );
      default:
        return null;
    }
  };

  return (
    <>
      <div className="max-w-screen-2xl mx-auto container px-
6 py-3 md:px-40 dark:text-white dark:bg-gray-900">
        <div className="flex min-h-screen items-center
justify-center pt-16 pb-8">
          <div className="border-2 border-dashed px-4 py-2
md:px-8 md:py-6 border-indigo-400 rounded-lg shadow-lg w-
full max-w-2xl">
            <h1 className="text-3xl font-bold text-center
mb-4">
              File Converter Online
            </h1>
            <p className="text-sm text-center mb-5">
              Easily convert files to any format online,
without having
              to install any software.
            </p>

            <div className="flex flex-col items-center
space-y-4">
              {!isConverted ? (
```

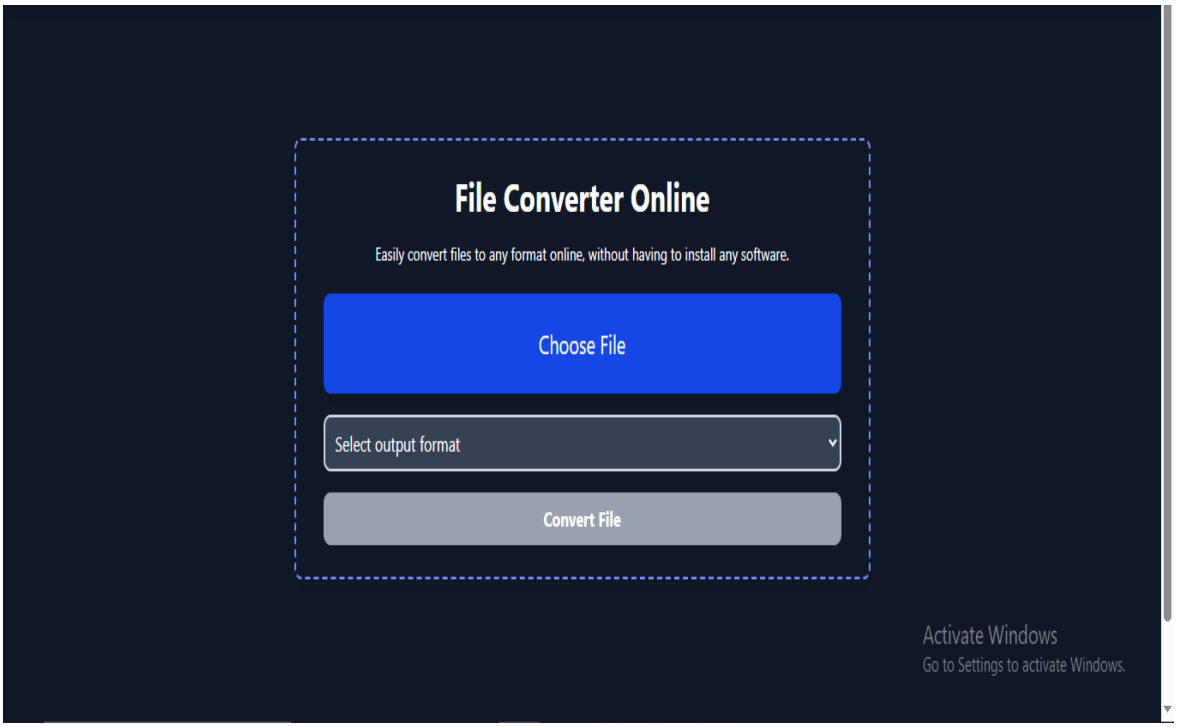
```
                <>
                  <input
                    type="file"
                    accept=".doc,.docx,.jpg,.pdf,.png"
                    onChange={handleFileChange}
                    className="hidden"
                    id="FileInput"
                  />
                  <label
                    htmlFor="FileInput"
                    className="w-full flex items-center
justify-center px-4 py-6 bg-gray-100 text-gray-700 rounded-
lg shadow-lg cursor-pointer border-blue-300 hover:bg-blue-
700 duration-300 hover:text-white"
                  >
                    {defaultFileIcon(selectedFile)}
                    <span className="text-xl">
                      {selectedFile ? selectedFile.name :
"Choose File"}
                    </span>
                  </label>

                  <select
                    className="w-full py-2 border-2 border-
gray-300 rounded-lg px-2 dark:bg-gray-700 dark:text-white
cursor-pointer"
                    value={conversionFormat}
                    onChange={handleConvertFormat}
                  >
                    <option className="cursor-pointer"
value="" disabled>
                      Select output format
                    </option>
                    <option className="cursor-
pointer"  value="docx"
disabled={selectedFile?.name?.split('.').pop().toLowerCase()
=== "docx"}>
                      DOCX
                    </option>
```

```
                    <option className="cursor-
pointer"  value="pdf"
disabled={selectedFile?.name?.split('.').pop().toLowerCase()
=== "pdf"}>
                      PDF
                    </option>
                    <option className="cursor-
pointer"  value="jpg"
disabled={selectedFile?.name?.split('.').pop().toLowerCase()
=== "jpg"}>
                      JPG
                    </option>
                    <option className="cursor-
pointer"  value="png"
disabled={selectedFile?.name?.split('.').pop().toLowerCase()
=== "png"}>
                      PNG
                    </option>
                  </select>

                  <button
                    onClick={handleConversion}
                    disabled={!selectedFile ||
!conversionFormat || loading}
                    className="w-full flex items-center
justify-center text-white bg-blue-500 hover:bg-blue-700
disabled:bg-gray-400 disabled:pointer-events-none duration-
300 font-bold px-4 py-2 rounded-lg cursor-pointer"
                  >
                    {loading ? (
                      <>
                        <div className="animate-spin
rounded-full h-5 w-5 border-b-2 border-white mr-2"></div>
                        Converting...

                      </>
                    ) : (
                      "Convert File"
                    )}
```

```
                  </button>
                </>
              ) : (
                <div className="w-full dark:bg-gray-800
dark:text-white">
                  <div className="bg-green-50 border border-
green-200 rounded-lg p-4 mb-4">
                    <div className="flex items-center mb-2">
                      {defaultFileIcon({ name:
`file.${conversionFormat}` })}
                      <span className="font-semibold">
                        {selectedFile.name.replace(/\.[^/.]+
$/, `.${conversionFormat}`)}
                      </span>
                    </div>
                    <p className="text-green-700 text-sm mb-
3">Conversion completed successfully!</p>

                    <div className="flex gap-3">
                      <button
                        onClick={handleDownload}
                        className="flex-1 flex items-center
justify-center text-white bg-green-500 hover:bg-green-700
font-bold px-4 py-2 rounded-lg cursor-pointer"
                      >
                        <FaDownload className="mr-2 " />
Download
                      </button>

                      <button
                        onClick={resetAll}
                        className="flex items-center
justify-center cursor-pointer text-indigo-600 bg-white
border border-indigo-600 hover:bg-indigo-50 font-bold px-4
py-2 rounded-lg"
                      >
                        <FaRedo className="mr-2 " /> New
Conversion
                      </button>
```

```
                    </div>
                  </div>

                  {renderPreview()}
                </div>
              )}

              {message && !isConverted && (
                <div className="text-green-500 text-
center">{message}</div>
              )}

              {error && (
                <div className="text-red-500 text-
center">{error}</div>
              )}
            </div>
          </div>
        </div>
      </>
    );
}

export default Home;
```

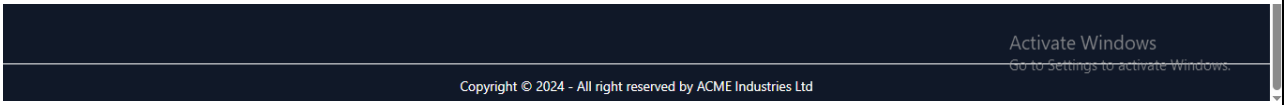
Home page


File Converter Online
Easily convert files to any format online, without having to install any software.
Choose File
Select output format
Convert File
Activate Windows
Go to Settings to activate Windows.

Footer Component Source Code

```
import React from "react";

function Footer() {
  return (
    <>
      <div className="items-center justify-center
border-t-1 text-black border-black dark:border-white
bg-white dark:text-white dark:bg-gray-900">

        <h1 className="text-center py-3 text-sm">
          Copyright © 2024 - All right reserved by
ACME Industries Ltd
        </h1>
      </div>
    </>
  );
}

export default Footer;
```

Footer Frontend



ThemeButton Component Source Code

This component is change to change the theme of The page.

```
import React from 'react'
import useTheme from '../contexts/theme.js';

export default function ThemeBtn() {

    const {themeMode,lightTheme,darkTheme}=useTheme();

     const onChangeBtn=(e)=>{
         const darkModeStatus=e.currentTarget.checked
         if(darkModeStatus){
            darkTheme()
         }
```

```
            else{
                lightTheme()
            }
        }
    return (
        <label className="relative inline-flex items-center
cursor-pointer">
            <input
                type="checkbox"
                value=""
                className="sr-only peer"
                onChange={onChangeBtn}
                checked={themeMode=== "dark"}
            />
            <div className="w-11 h-6 bg-gray-200 peer-
focus:outline-none peer-focus:ring-4 peer-focus:ring-blue-
300 dark:peer-focus:ring-blue-800 rounded-full peer dark:bg-
gray-700 peer-checked:after:translate-x-full peer-
checked:after:border-white after:content-[''] after:absolute
after:top-[2px] after:left-[2px] after:bg-white
after:border-gray-300 after:border after:rounded-full
after:h-5 after:w-5 after:transition-all dark:border-gray-
600 peer-checked:bg-blue-600"></div>
            <span className="ml-3 text-m font-medium text-
black dark:text-white">Toggle Theme</span>
        </label>
    );
}
```

ThemeButton Component

Contact Form Source Code

```
import React from "react";

const Contact = () => {
  return (
    <>
      <section className="relative z-10 overflow-hidden  bg-
white dark:text-white  py-20 dark:bg-black lg:py-[120px]">
        <div className="container">
          <div className="-mx-4 flex flex-wrap lg:justify-
between">
            <div className="w-full px-4 lg:w-1/2 xl:w-6/12">
              <div className="mb-12 max-w-[570px] lg:mb-0">
                <span className="mb-4 ml-2 block text-base
font-semibold text-primary">
                  Contact Us
                </span>
                <h2 className="mb-6 ml-2 text-[32px] font-
bold uppercase text-dark dark:text-white sm:text-[40px]
lg:text-[36px] xl:text-[40px]">
                  GET IN TOUCH WITH US
                </h2>
```

```
                <p className="mb-9  ml-2 text-base leading-
relaxed text-body-color dark:text-dark-6">
                We’d love to hear from you! Whether you have
a question, feedback, or a project in mind, feel free to get
in touch with us.
                </p>
                <div className="mb-8 flex w-full max-w-
[370px]">
                  <div className="mr-6 flex h-[60px] w-full
max-w-[60px] items-center justify-center overflow-hidden
rounded bg-primary/5 text-primary sm:h-[70px] sm:max-w-
[70px]">
                    <svg
                      width="32"
                      height="32"
                      viewBox="0 0 32 32"
                      fill="none"
                      xmlns="http://www.w3.org/2000/svg"
                    >
                      <path
                        d="M30.6 11.8002L17.7 3.5002C16.65
2.8502 15.3 2.8502 14.3 3.5002L1.39998 11.8002C0.899983
12.1502 0.749983 12.8502 1.04998 13.3502C1.39998 13.8502
2.09998 14.0002 2.59998 13.7002L3.44998
13.1502V25.8002C3.44998 27.5502 4.84998 28.9502 6.59998
28.9502H25.4C27.15 28.9502 28.55 27.5502 28.55
25.8002V13.1502L29.4 13.7002C29.6 13.8002 29.8 13.9002 30
13.9002C30.35 13.9002 30.75 13.7002 30.95 13.4002C31.3
12.8502 31.15 12.1502 30.6 11.8002ZM13.35
26.7502V18.5002C13.35 18.0002 13.75 17.6002 14.25
17.6002H17.75C18.25 17.6002 18.65 18.0002 18.65
18.5002V26.7502H13.35ZM26.3 25.8002C26.3 26.3002 25.9
26.7002 25.4 26.7002H20.9V18.5002C20.9 16.8002 19.5 15.4002
17.8 15.4002H14.3C12.6 15.4002 11.2 16.8002 11.2
18.5002V26.7502H6.69998C6.19998 26.7502 5.79998 26.3502
5.79998 25.8502V11.7002L15.5 5.4002C15.8 5.2002 16.2 5.2002
16.5 5.4002L26.3 11.7002V25.8002Z"
                        fill="currentColor"
                      />
```

```
                    </svg>
                  </div>
                  <div className="w-full">
                    <h4 className="mb-1 text-xl font-bold
text-dark dark:text-white">
                      Our Location
                    </h4>
                    <p className="text-base text-body-color
dark:text-dark-6">
                      Gorakhpur ,Uttar Pradesh
                    </p>
                  </div>
                </div>

                <div className="mb-8 flex w-full max-w-
[370px]">
                  <div className="mr-6 flex h-[60px] w-full
max-w-[60px] items-center justify-center overflow-hidden
rounded bg-primary/5 text-primary sm:h-[70px] sm:max-w-
[70px]">
                    <svg
                      width="32"
                      height="32"
                      viewBox="0 0 32 32"
                      fill="none"
                      xmlns="http://www.w3.org/2000/svg"
                    >
                      <g clip-path="url(#clip0_941_17577)">
                        <path
                          d="M24.3 31.1499C22.95 31.1499
21.4 30.7999 19.7 30.1499C16.3 28.7999 12.55 26.1999 9.19997
22.8499C5.84997 19.4999 3.24997 15.7499 1.89997
12.2999C0.39997 8.59994 0.54997 5.54994 2.29997
3.84994C2.34997 3.79994 2.44997 3.74994 2.49997
3.69994L6.69997 1.19994C7.74997 0.599942 9.09997 0.899942
9.79997 1.89994L12.75 6.29994C13.45 7.34994 13.15 8.74994
12.15 9.44994L10.35 10.6999C11.65 12.7999 15.35 17.9499
21.25 21.6499L22.35 20.0499C23.2 18.8499 24.55 18.4999 25.65
19.2499L30.05 22.1999C31.05 22.8999 31.35 24.2499 30.75
```

```
25.2999L28.25 29.4999C28.2 29.5999 28.15 29.6499 28.1
29.6999C27.2 30.6499 25.9 31.1499 24.3 31.1499ZM3.79997
5.54994C2.84997 6.59994 2.89997 8.74994 3.99997
11.4999C5.24997 14.6499 7.64997 18.0999 10.8 21.2499C13.9
24.3499 17.4 26.7499 20.5 27.9999C23.2 29.0999 25.35 29.1499
26.45 28.1999L28.85 24.0999C28.85 24.0499 28.85 24.0499
28.85 23.9999L24.45 21.0499C24.45 21.0499 24.35 21.0999
24.25 21.2499L23.15 22.8499C22.45 23.8499 21.1 24.1499 20.1
23.4999C13.8 19.5999 9.89997 14.1499 8.49997 11.9499C7.84997
10.8999 8.09997 9.54994 9.09997 8.84994L10.9
7.59994V7.54994L7.94997 3.14994C7.94997 3.09994 7.89997
3.09994 7.84997 3.14994L3.79997 5.54994Z"
                          fill="currentColor"
                        />
                        <path
                          d="M29.3 14.25C28.7 14.25 28.25
13.8 28.2 13.2C27.8 8.15003 23.65 4.10003 18.55
3.75003C17.95 3.70003 17.45 3.20003 17.5 2.55003C17.55
1.95003 18.05 1.45003 18.7 1.50003C24.9 1.90003 29.95
6.80003 30.45 13C30.5 13.6 30.05 14.15 29.4 14.2C29.4 14.25
29.35 14.25 29.3 14.25Z"
                          fill="currentColor"
                        />
                        <path
                          d="M24.35 14.7002C23.8 14.7002
23.3 14.3002 23.25 13.7002C22.95 11.0002 20.85 8.90018 18.15
8.55018C17.55 8.50018 17.1 7.90018 17.15 7.30018C17.2
6.70018 17.8 6.25018 18.4 6.30018C22.15 6.75018 25.05
9.65018 25.5 13.4002C25.55 14.0002 25.15 14.5502 24.5
14.6502C24.4 14.7002 24.35 14.7002 24.35 14.7002Z"
                          fill="currentColor"
                        />
                      </g>
                      <defs>
                        <clipPath id="clip0_941_17577">
                          <rect width="32" height="32"
fill="white" />
                        </clipPath>
                      </defs>
```

```
                    </svg>
                  </div>
                  <div className="w-full">
                    <h4 className="mb-1 text-xl font-bold
text-dark dark:text-white">
                      Phone Number
                    </h4>
                    <p className="text-base text-body-color
dark:text-dark-6">
                      (+91) 123456789
                    </p>
                  </div>
                </div>

                <div className="mb-8 flex w-full max-w-
[370px]">
                  <div className="mr-6 flex h-[60px] w-full
max-w-[60px] items-center justify-center overflow-hidden
rounded bg-primary/5 text-primary sm:h-[70px] sm:max-w-
[70px]">
                    <svg
                      width="32"
                      height="32"
                      viewBox="0 0 32 32"
                      fill="none"
                      xmlns="http://www.w3.org/2000/svg"
                    >
                      <path
                        d="M28 4.7998H3.99998C2.29998 4.7998
0.849976 6.1998 0.849976 7.9498V24.1498C0.849976 25.8498
2.24998 27.2998 3.99998 27.2998H28C29.7 27.2998 31.15
25.8998 31.15 24.1498V7.8998C31.15 6.1998 29.7 4.7998 28
4.7998ZM28 7.0498C28.05 7.0498 28.1 7.0498 28.15 7.0498L16
14.8498L3.84998 7.0498C3.89998 7.0498 3.94998 7.0498 3.99998
7.0498H28ZM28 24.9498H3.99998C3.49998 24.9498 3.09998
24.5498 3.09998 24.0498V9.2498L14.8 16.7498C15.15 16.9998
15.55 17.0998 15.95 17.0998C16.35 17.0998 16.75 16.9998 17.1
16.7498L28.8 9.2498V24.0998C28.9 24.5998 28.5 24.9498 28
24.9498Z"
```

```
                        fill="currentColor"
                      />
                    </svg>
                  </div>
                  <div className="w-full">
                    <h4 className="mb-1 text-xl font-bold
text-dark dark:text-white">
                      Email Address
                    </h4>
                    <p className="text-base text-body-color
dark:text-dark-6">
                     fileconversion@gmail.com
                    </p>
                  </div>
                </div>
              </div>
            </div>
            <div className="w-full px-4 lg:w-1/2 xl:w-5/12">
              <div className="relative rounded-lg bg-white
p-8 shadow-lg dark:bg-dark-2 sm:p-12 dark:text-white
dark:bg-black">
                <form>
                  <ContactInputBox
                    type="text"
                    name="name"
                    placeholder="Your Name"
                  />
                  <ContactInputBox
                    type="text"
                    name="email"
                    placeholder="Your Email"
                  />
                  <ContactInputBox
                    type="text"
                    name="phone"
                    placeholder="Your Phone"
                  />
                  <ContactTextArea
                    row="6"
```

```
                    placeholder="Your Message"
                    name="details"
                    defaultValue=""
                  />
                  <div>
                    <button
                      type="submit"
                      className="w-full rounded border
border-primary bg-primary p-3 bg-black text-white transition
hover:bg-opacity-90 dark:text-black dark:bg-white"
                    >
                      Send Message
                    </button>
                  </div>
                </form>
                <div>
                  <span className="absolute -right-9 -top-10
z-[-1]">
                    <svg
                      width={100}
                      height={100}
                      viewBox="0 0 100 100"
                      fill="none"
                      xmlns="http://www.w3.org/2000/svg"
                    >
                      <path
                        fillRule="evenodd"
                        clipRule="evenodd"
                        d="M0 100C0 44.7715 0 0 0 0C55.2285
0 100 44.7715 100 100C100 100 100 100 0 100Z"
                        fill="#3056D3"
                      />
                    </svg>
                  </span>
                  <span className="absolute -right-10 top-
[90px] z-[-1]">
                    <svg
                      width={34}
                      height={134}
```

```
                    viewBox="0 0 34 134"
                    fill="none"
                    xmlns="http://www.w3.org/2000/svg"
                  >
                    <circle
                      cx="31.9993"
                      cy={132}
                      r="1.66667"
                      transform="rotate(180 31.9993 132)"
                      fill="#13C296"
                    />
                    <circle
                      cx="31.9993"
                      cy="117.333"
                      r="1.66667"
                      transform="rotate(180 31.9993
117.333)"
                      fill="#13C296"
                    />
                    <circle
                      cx="31.9993"
                      cy="102.667"
                      r="1.66667"
                      transform="rotate(180 31.9993
102.667)"
                      fill="#13C296"
                    />
                    <circle
                      cx="31.9993"
                      cy={88}
                      r="1.66667"
                      transform="rotate(180 31.9993 88)"
                      fill="#13C296"
                    />
                    <circle
                      cx="31.9993"
                      cy="73.3333"
                      r="1.66667"
```

```
                    transform="rotate(180 31.9993
73.3333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy={45}
                    r="1.66667"
                    transform="rotate(180 31.9993 45)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy={16}
                    r="1.66667"
                    transform="rotate(180 31.9993 16)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy={59}
                    r="1.66667"
                    transform="rotate(180 31.9993 59)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy="30.6666"
                    r="1.66667"
                    transform="rotate(180 31.9993
30.6666)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy="1.66665"
                    r="1.66667"
                    transform="rotate(180 31.9993
1.66665)"
```

```
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy={132}
                    r="1.66667"
                    transform="rotate(180 17.3333 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 17.3333
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy="102.667"
                    r="1.66667"
                    transform="rotate(180 17.3333
102.667)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy={88}
                    r="1.66667"
                    transform="rotate(180 17.3333 88)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy="73.3333"
                    r="1.66667"
                    transform="rotate(180 17.3333
73.3333)"
                    fill="#13C296"
```

```
                    />
                    <circle
                      cx="17.3333"
                      cy={45}
                      r="1.66667"
                      transform="rotate(180 17.3333 45)"
                      fill="#13C296"
                    />
                    <circle
                      cx="17.3333"
                      cy={16}
                      r="1.66667"
                      transform="rotate(180 17.3333 16)"
                      fill="#13C296"
                    />
                    <circle
                      cx="17.3333"
                      cy={59}
                      r="1.66667"
                      transform="rotate(180 17.3333 59)"
                      fill="#13C296"
                    />
                    <circle
                      cx="17.3333"
                      cy="30.6666"
                      r="1.66667"
                      transform="rotate(180 17.3333
30.6666)"
                      fill="#13C296"
                    />
                    <circle
                      cx="17.3333"
                      cy="1.66665"
                      r="1.66667"
                      transform="rotate(180 17.3333
1.66665)"
                      fill="#13C296"
                    />
                    <circle
```

```
                    cx="2.66536"
                    cy={132}
                    r="1.66667"
                    transform="rotate(180 2.66536 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 2.66536
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy="102.667"
                    r="1.66667"
                    transform="rotate(180 2.66536
102.667)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy={88}
                    r="1.66667"
                    transform="rotate(180 2.66536 88)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy="73.3333"
                    r="1.66667"
                    transform="rotate(180 2.66536
73.3333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
```

```
                    cy={45}
                    r="1.66667"
                    transform="rotate(180 2.66536 45)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy={16}
                    r="1.66667"
                    transform="rotate(180 2.66536 16)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy={59}
                    r="1.66667"
                    transform="rotate(180 2.66536 59)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy="30.6666"
                    r="1.66667"
                    transform="rotate(180 2.66536
30.6666)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy="1.66665"
                    r="1.66667"
                    transform="rotate(180 2.66536
1.66665)"
                    fill="#13C296"
                  />
                </svg>
              </span>
              <span className="absolute -bottom-7 -left-
7 z-[-1]">
```

```
                    <svg
                      width={107}
                      height={134}
                      viewBox="0 0 107 134"
                      fill="none"
                      xmlns="http://www.w3.org/2000/svg"
                    >
                      <circle
                        cx="104.999"
                        cy={132}
                        r="1.66667"
                        transform="rotate(180 104.999 132)"
                        fill="#13C296"
                      />
                      <circle
                        cx="104.999"
                        cy="117.333"
                        r="1.66667"
                        transform="rotate(180 104.999
117.333)"
                        fill="#13C296"
                      />
                      <circle
                        cx="104.999"
                        cy="102.667"
                        r="1.66667"
                        transform="rotate(180 104.999
102.667)"
                        fill="#13C296"
                      />
                      <circle
                        cx="104.999"
                        cy={88}
                        r="1.66667"
                        transform="rotate(180 104.999 88)"
                        fill="#13C296"
                      />
                      <circle
                        cx="104.999"
```

```
                    cy="73.3333"
                    r="1.66667"
                    transform="rotate(180 104.999
73.3333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="104.999"
                    cy={45}
                    r="1.66667"
                    transform="rotate(180 104.999 45)"
                    fill="#13C296"
                  />
                  <circle
                    cx="104.999"
                    cy={16}
                    r="1.66667"
                    transform="rotate(180 104.999 16)"
                    fill="#13C296"
                  />
                  <circle
                    cx="104.999"
                    cy={59}
                    r="1.66667"
                    transform="rotate(180 104.999 59)"
                    fill="#13C296"
                  />
                  <circle
                    cx="104.999"
                    cy="30.6666"
                    r="1.66667"
                    transform="rotate(180 104.999
30.6666)"
                    fill="#13C296"
                  />
                  <circle
                    cx="104.999"
                    cy="1.66665"
                    r="1.66667"
```

```
                    transform="rotate(180 104.999
1.66665)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy={132}
                    r="1.66667"
                    transform="rotate(180 90.3333 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 90.3333
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy="102.667"
                    r="1.66667"
                    transform="rotate(180 90.3333
102.667)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy={88}
                    r="1.66667"
                    transform="rotate(180 90.3333 88)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy="73.3333"
                    r="1.66667"
```

```
                    transform="rotate(180 90.3333
73.3333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy={45}
                    r="1.66667"
                    transform="rotate(180 90.3333 45)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy={16}
                    r="1.66667"
                    transform="rotate(180 90.3333 16)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy={59}
                    r="1.66667"
                    transform="rotate(180 90.3333 59)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy="30.6666"
                    r="1.66667"
                    transform="rotate(180 90.3333
30.6666)"
                    fill="#13C296"
                  />
                  <circle
                    cx="90.3333"
                    cy="1.66665"
                    r="1.66667"
                    transform="rotate(180 90.3333
1.66665)"
```

```
                    fill="#13C296"
                  />
                  <circle
                    cx="75.6654"
                    cy={132}
                    r="1.66667"
                    transform="rotate(180 75.6654 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy={132}
                    r="1.66667"
                    transform="rotate(180 31.9993 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="75.6654"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 75.6654
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 31.9993
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="75.6654"
                    cy="102.667"
                    r="1.66667"
                    transform="rotate(180 75.6654
102.667)"
                    fill="#13C296"
```

```
                    />
                    <circle
                      cx="31.9993"
                      cy="102.667"
                      r="1.66667"
                      transform="rotate(180 31.9993
102.667)"
                      fill="#13C296"
                    />
                    <circle
                      cx="75.6654"
                      cy={88}
                      r="1.66667"
                      transform="rotate(180 75.6654 88)"
                      fill="#13C296"
                    />
                    <circle
                      cx="31.9993"
                      cy={88}
                      r="1.66667"
                      transform="rotate(180 31.9993 88)"
                      fill="#13C296"
                    />
                    <circle
                      cx="75.6654"
                      cy="73.3333"
                      r="1.66667"
                      transform="rotate(180 75.6654
73.3333)"
                      fill="#13C296"
                    />
                    <circle
                      cx="31.9993"
                      cy="73.3333"
                      r="1.66667"
                      transform="rotate(180 31.9993
73.3333)"
                      fill="#13C296"
                    />
```

```
<circle
  cx="75.6654"
  cy={45}
  r="1.66667"
  transform="rotate(180 75.6654 45)"
  fill="#13C296"
/>
<circle
  cx="31.9993"
  cy={45}
  r="1.66667"
  transform="rotate(180 31.9993 45)"
  fill="#13C296"
/>
<circle
  cx="75.6654"
  cy={16}
  r="1.66667"
  transform="rotate(180 75.6654 16)"
  fill="#13C296"
/>
<circle
  cx="31.9993"
  cy={16}
  r="1.66667"
  transform="rotate(180 31.9993 16)"
  fill="#13C296"
/>
<circle
  cx="75.6654"
  cy={59}
  r="1.66667"
  transform="rotate(180 75.6654 59)"
  fill="#13C296"
/>
<circle
  cx="31.9993"
  cy={59}
  r="1.66667"
```

```
                    transform="rotate(180 31.9993 59)"
                    fill="#13C296"
                  />
                  <circle
                    cx="75.6654"
                    cy="30.6666"
                    r="1.66667"
                    transform="rotate(180 75.6654
30.6666)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy="30.6666"
                    r="1.66667"
                    transform="rotate(180 31.9993
30.6666)"
                    fill="#13C296"
                  />
                  <circle
                    cx="75.6654"
                    cy="1.66665"
                    r="1.66667"
                    transform="rotate(180 75.6654
1.66665)"
                    fill="#13C296"
                  />
                  <circle
                    cx="31.9993"
                    cy="1.66665"
                    r="1.66667"
                    transform="rotate(180 31.9993
1.66665)"
                    fill="#13C296"
                  />
                  <circle
                    cx="60.9993"
                    cy={132}
                    r="1.66667"
```

```
                    transform="rotate(180 60.9993 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy={132}
                    r="1.66667"
                    transform="rotate(180 17.3333 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="60.9993"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 60.9993
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 17.3333
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="60.9993"
                    cy="102.667"
                    r="1.66667"
                    transform="rotate(180 60.9993
102.667)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy="102.667"
                    r="1.66667"
```

```
                    transform="rotate(180 17.3333
102.667)"
                    fill="#13C296"
                  />
                  <circle
                    cx="60.9993"
                    cy={88}
                    r="1.66667"
                    transform="rotate(180 60.9993 88)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy={88}
                    r="1.66667"
                    transform="rotate(180 17.3333 88)"
                    fill="#13C296"
                  />
                  <circle
                    cx="60.9993"
                    cy="73.3333"
                    r="1.66667"
                    transform="rotate(180 60.9993
73.3333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="17.3333"
                    cy="73.3333"
                    r="1.66667"
                    transform="rotate(180 17.3333
73.3333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="60.9993"
                    cy={45}
                    r="1.66667"
                    transform="rotate(180 60.9993 45)"
```

```
    fill="#13C296"
  />
  <circle
    cx="17.3333"
    cy={45}
    r="1.66667"
    transform="rotate(180 17.3333 45)"
    fill="#13C296"
  />
  <circle
    cx="60.9993"
    cy={16}
    r="1.66667"
    transform="rotate(180 60.9993 16)"
    fill="#13C296"
  />
  <circle
    cx="17.3333"
    cy={16}
    r="1.66667"
    transform="rotate(180 17.3333 16)"
    fill="#13C296"
  />
  <circle
    cx="60.9993"
    cy={59}
    r="1.66667"
    transform="rotate(180 60.9993 59)"
    fill="#13C296"
  />
  <circle
    cx="17.3333"
    cy={59}
    r="1.66667"
    transform="rotate(180 17.3333 59)"
    fill="#13C296"
  />
  <circle
    cx="60.9993"
```

```
                        cy="30.6666"
                        r="1.66667"
                        transform="rotate(180 60.9993
30.6666)"
                        fill="#13C296"
                      />
                      <circle
                        cx="17.3333"
                        cy="30.6666"
                        r="1.66667"
                        transform="rotate(180 17.3333
30.6666)"
                        fill="#13C296"
                      />
                      <circle
                        cx="60.9993"
                        cy="1.66665"
                        r="1.66667"
                        transform="rotate(180 60.9993
1.66665)"
                        fill="#13C296"
                      />
                      <circle
                        cx="17.3333"
                        cy="1.66665"
                        r="1.66667"
                        transform="rotate(180 17.3333
1.66665)"
                        fill="#13C296"
                      />
                      <circle
                        cx="46.3333"
                        cy={132}
                        r="1.66667"
                        transform="rotate(180 46.3333 132)"
                        fill="#13C296"
                      />
                      <circle
                        cx="2.66536"
```

```
                    cy={132}
                    r="1.66667"
                    transform="rotate(180 2.66536 132)"
                    fill="#13C296"
                  />
                  <circle
                    cx="46.3333"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 46.3333
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy="117.333"
                    r="1.66667"
                    transform="rotate(180 2.66536
117.333)"
                    fill="#13C296"
                  />
                  <circle
                    cx="46.3333"
                    cy="102.667"
                    r="1.66667"
                    transform="rotate(180 46.3333
102.667)"
                    fill="#13C296"
                  />
                  <circle
                    cx="2.66536"
                    cy="102.667"
                    r="1.66667"
                    transform="rotate(180 2.66536
102.667)"
                    fill="#13C296"
                  />
                  <circle
                    cx="46.3333"
```

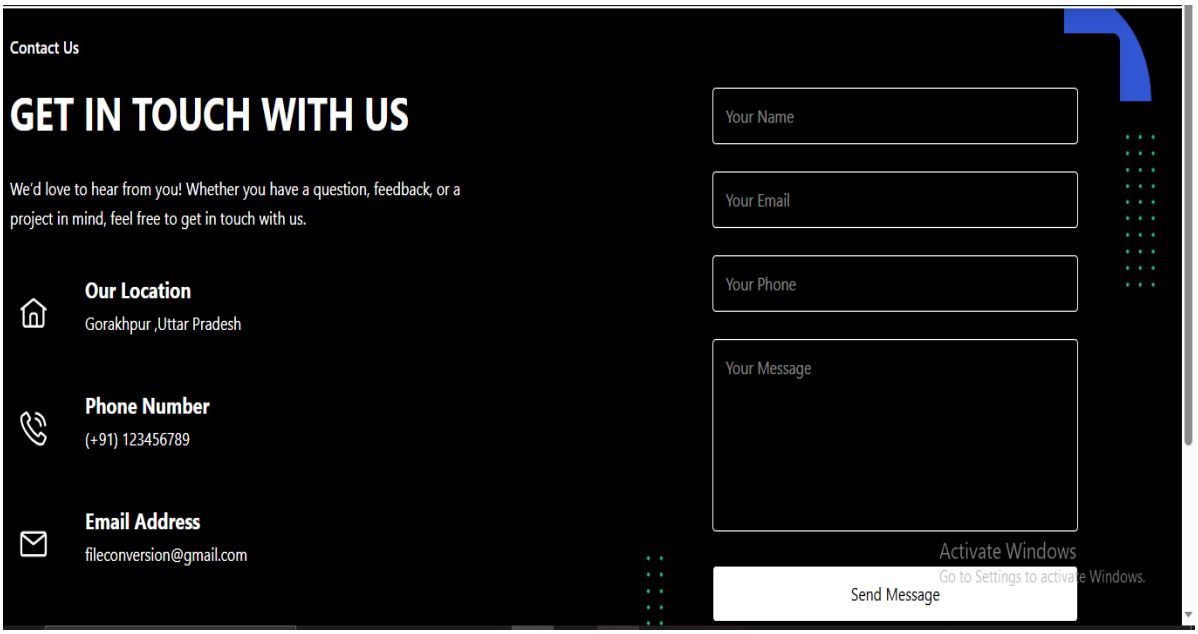
```
                        cy={88}
                        r="1.66667"
                        transform="rotate(180 46.3333 88)"
                        fill="#13C296"
                      />
                      <circle
                        cx="2.66536"
                        cy={88}
                        r="1.66667"
                        transform="rotate(180 2.66536 88)"
                        fill="#13C296"
                      />
                      <circle
                        cx="46.3333"
                        cy="73.3333"
                        r="1.66667"
                        transform="rotate(180 46.3333
73.3333)"
                        fill="#13C296"
                      />
                      <circle
                        cx="2.66536"
                        cy="73.3333"
                        r="1.66667"
                        transform="rotate(180 2.66536
73.3333)"
                        fill="#13C296"
                      />
                      <circle
                        cx="46.3333"
                        cy={45}
                        r="1.66667"
                        transform="rotate(180 46.3333 45)"
                        fill="#13C296"
                      />
                      <circle
                        cx="2.66536"
                        cy={45}
                        r="1.66667"
```

```
                transform="rotate(180 2.66536 45)"
                fill="#13C296"
              />
              <circle
                cx="46.3333"
                cy={16}
                r="1.66667"
                transform="rotate(180 46.3333 16)"
                fill="#13C296"
              />
              <circle
                cx="2.66536"
                cy={16}
                r="1.66667"
                transform="rotate(180 2.66536 16)"
                fill="#13C296"
              />
              <circle
                cx="46.3333"
                cy={59}
                r="1.66667"
                transform="rotate(180 46.3333 59)"
                fill="#13C296"
              />
              <circle
                cx="2.66536"
                cy={59}
                r="1.66667"
                transform="rotate(180 2.66536 59)"
                fill="#13C296"
              />
              <circle
                cx="46.3333"
                cy="30.6666"
                r="1.66667"
                transform="rotate(180 46.3333
30.6666)"
                fill="#13C296"
              />
```

```
                    <circle
                      cx="2.66536"
                      cy="30.6666"
                      r="1.66667"
                      transform="rotate(180 2.66536
30.6666)"
                      fill="#13C296"
                    />
                    <circle
                      cx="46.3333"
                      cy="1.66665"
                      r="1.66667"
                      transform="rotate(180 46.3333
1.66665)"
                      fill="#13C296"
                    />
                    <circle
                      cx="2.66536"
                      cy="1.66665"
                      r="1.66667"
                      transform="rotate(180 2.66536
1.66665)"
                      fill="#13C296"
                    />
                  </svg>
                </span>
              </div>
            </div>
          </div>
        </div>
      </section>
    </>
  );
};

export default Contact;
```

```jsx
const ContactTextArea = ({ row, placeholder, name,
defaultValue }) => {
  return (
    <>
      <div className="mb-6">
        <textarea
          rows={row}
          placeholder={placeholder}
          name={name}
          className="w-full resize-none rounded border
border-stroke px-[14px] py-3 text-base text-body-color
outline-none focus:border-primary dark:border-dark-3
dark:bg-dark dark:text-dark-6"
          defaultValue={defaultValue}
        />
      </div>
    </>
  );
};

const ContactInputBox = ({ type, placeholder, name }) => {
  return (
    <>
      <div className="mb-6">
        <input
          type={type}
          placeholder={placeholder}
          name={name}
          className="w-full rounded border border-stroke px-
[14px] py-3 text-base text-body-color outline-none
focus:border-primary dark:border-dark-3 dark:bg-dark
dark:text-dark-6"
        />
      </div>
    </>
  );
};
```

Contact Page

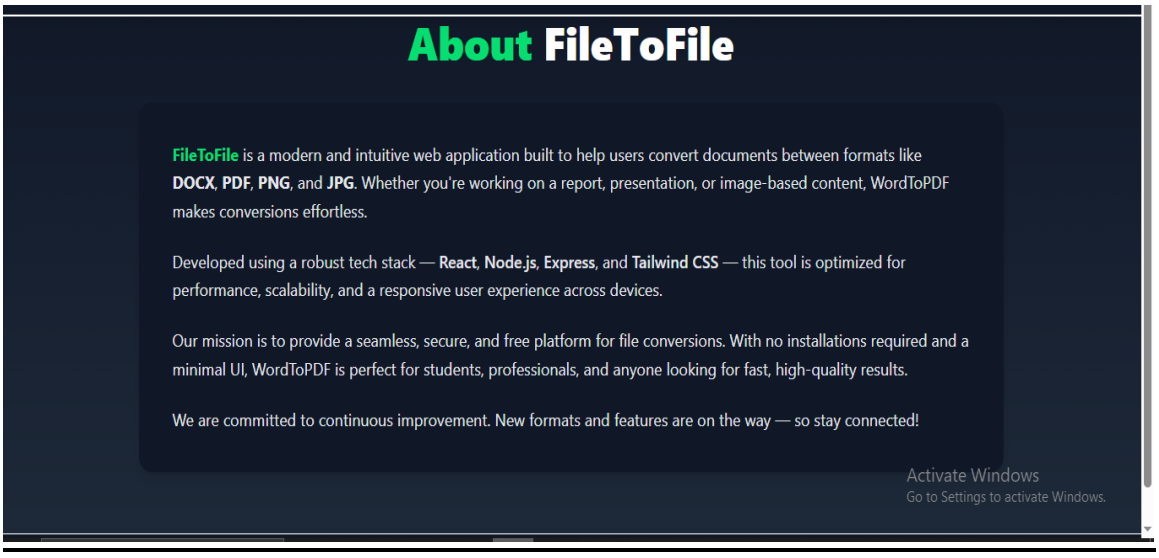


About Component Source Code

```
import React from 'react';

const About = () => {
  return (
    <section className="relative py-16 px-6 sm:px-10 bg-gradient-to-b
from-white to-gray-100 dark:from-gray-900 dark:to-gray-800 text-gray-
800 dark:text-gray-200">
      <div className="max-w-5xl mx-auto">
        <h1 className="text-4xl sm:text-5xl font-extrabold text-center
mb-10 text-green-600 dark:text-green-400">
          About <span className="text-gray-900 dark:text-
white">FileToFile</span>
        </h1>

        <div className="bg-white dark:bg-gray-900 rounded-2xl shadow-
lg p-8 sm:p-10 space-y-6">
          <p className="text-lg leading-relaxed">
            <strong className="text-green-600 dark:text-green-
400">FileToFile</strong> is a modern and intuitive web application
built to help users convert documents between formats like
<strong>DOCX</strong>, <strong>PDF</strong>, <strong>PNG</strong>, and
<strong>JPG</strong>. Whether you're working on a report,
presentation, or image-based content, WordToPDF makes conversions
effortless.
          </p>

          <p className="text-lg leading-relaxed">
            Developed using a robust tech stack — <span
className="font-semibold">React</span>, <span className="font-
semibold">Node.js</span>, <span className="font-
semibold">Express</span>, and <span className="font-semibold">Tailwind
CSS</span> — this tool is optimized for performance, scalability, and
a responsive user experience across devices.
          </p>

          <p className="text-lg leading-relaxed">
            Our mission is to provide a seamless, secure, and free
platform for file conversions. With no installations required and a
minimal UI, WordToPDF is perfect for students, professionals, and
anyone looking for fast, high-quality results.
```

```
          </p>

          <p className="text-lg leading-relaxed">
            We are committed to continuous improvement. New formats
and features are on the way — so stay connected!
          </p>
        </div>
      </div>
    </section>
  );
};

export default About;
```

About Page



Frontend


FileToFile
Home
Contact
About
Toggle Theme
File Converter Online
Easily convert files to any format online, without having to install any software.
Choose File
Select output format
Convert File
Activate Windows
Go to Settings to activate Windows.
Copyright © 2024 - All right reserved by ACME Industries Ltd

Backend

index.js

```
const express=require('express');
const app=express();
const multer=require('multer');
const path=require('path');
const cors=require('cors');
const axios=require('axios');
const  cleanFolder=require('./utils/CleanUploadFolder');
const archiver = require('archiver');

app.use(cors());
const port=3001|| process.env.PORT;
const fs = require('fs');
require('dotenv').config();

const convertapi =
require('convertapi')(process.env.CONVERT_API_SECRET);


const { Document, Packer, Paragraph, ImageRun } =
require("docx");

const FormData = require("form-data");



    const storage = multer.diskStorage({
        limits: { fileSize: 2 * 1024 * 1024 },
        destination: "uploads",
```

```
        filename: (req, file, cb) => {
          cb(null, file.originalname);  // Keep the original
filename
        }
      });


      const uploads=multer({storage:storage});



 app.post('/convertFile/docx-to-pdf',
uploads.single('file'), async(req,res,next)=>{
        try{
        if(!req.file){
            res.status(400).send('Please upload a file');
        }
        const docxfile=req.file.path;
        console.log(docxfile);
        const outputdir="downloads";
        const outputfile=path.join(outputdir,
`${path.basename(docxfile,".docx")}.pdf`);
        console.log(outputfile);

        const result= await convertapi.convert('pdf',
{File:docxfile}, 'docx');
         await result.saveFiles(outputfile);


         res.download(outputfile,(err)=>{
            if(err){
                console.log("error occurred",err);
            }
            console.log("file downloaded");
            cleanFolder("uploads");

         })

    }catch(err){
```

```
        console.log("error occurred",err);
    }

 })



app.post('/convertFile/docx-to-jpg', uploads.single("file"),
async (req, res, next) => {
    try {
        if (!req.file) {
            return res.status(400).json({ message: "No file
uploaded" });
        }
        const docxPath = req.file.path;
        const outputdir="downloads";
        const outputfile = path.join(outputdir,
path.basename(docxPath, ".docx"));

        // Create directory for output files if it doesn't
exist


        // Convert DOCX to JPG using convertapi
        const result = await convertapi.convert("jpg", {
            File: docxPath,
            // Set ImageQuality if needed (optional)
            ImageQuality: 90,
            // Specify to create separate JPGs (one per
page)
            StoreFile: true
        }, "docx");

        // Save all converted files to the output directory
        await result.saveFiles(outputfile);

        // Create a zip file containing all JPG files
        const zipPath = `${outputfile}.zip`;
```

```
        const archive = archiver('zip', { zlib: { level: 9 }
});
        const output = fs.createWriteStream(zipPath);

        // Wait for the zip to finish
        await new Promise((resolve, reject) => {
            output.on('close', resolve);
            archive.on('error', reject);
            archive.pipe(output);
            archive.directory(outputfile, false);
            archive.finalize();
        });

        // Send zip file containing all JPG images
        res.download(zipPath, `${path.basename(docxPath,
".docx")}.zip`, (err) => {
            if (err) {
                console.error("Error sending file:", err);
            } else {
                console.log("Files downloaded
successfully");
                // Clean up
                cleanFolder("uploads");



    }
});
} catch (error) {
        console.error("Error occurred:", error);
        res.status(500).json({ message: "Internal server
error" });
    }
});
```

```
app.post('/convertFile/docx-to-png', uploads.single("file"),
async (req, res) => {
    try {
        if (!req.file) {
            return res.status(400).json({ message: "No file
uploaded" });
        }

        const docxPath = req.file.path;
        const outputDir = "downloads";
        const {name}=path.parse(docxPath);
        // const outputFile = path.join(outputDir,
`${path.basename(docxPath, ".docx")}.png`); //for lowercase
          const
outputFile=path.join(outputDir,`${name}.png`)
         const result=await convertapi.convert("png",
{File:docxPath}, "docx");
            await result.saveFiles(outputFile);

        // Download the first converted image (modify if
multiple pages)
        res.download(outputFile, (err) => {
            if (err) {
                console.error("Error sending file:", err);
            } else {
                console.log("File downloaded successfully");
                cleanFolder("uploads");
            }
        });

    } catch (error) {
        console.error("Error occurred:", error);
        res.status(500).json({ message: "Internal server
error" });
    }
});

 app.post('/convertFile/pdf-to-docx',
uploads.single('file'), async(req,res,next)=>{
```

```
        try{
        if(!req.file){
            res.status(400).send('Please upload a file');
        }
        const pdfFile=req.file.path;
        console.log(pdfFile);
        const outputdir="downloads";
        const outputfile=path.join(outputdir,
`${path.basename(pdfFile,".pdf")}.docx`);
        console.log(outputfile);

        const result= await convertapi.convert('docx',
{File:pdfFile}, 'pdf');
         await result.saveFiles(outputfile);
        // await result.saveFiles(outputfile);

         res.download(outputfile,(err)=>{
            if(err){
                console.log("error occurred",err);
            }
            console.log("file downloaded");
            cleanFolder("uploads");
         })

    }catch(err){
        console.log("error occurred",err);
    }

 })

 app.post('/convertFile/pdf-to-jpg', uploads.single('file'),
async(req,res,next)=>{
        try{
        if(!req.file){
            res.status(400).send('Please upload a file');
        }
        const pdfFile=req.file.path;
        console.log(pdfFile);
        const outputdir="downloads";
```

```
        const outputfile=path.join(outputdir,
`${path.basename(pdfFile,".pdf")}.jpg`);
        console.log(outputfile);

        const result= await convertapi.convert('jpg',
{File:pdfFile}, 'pdf');
         await result.saveFiles(outputfile);


         res.download(outputfile,(err)=>{
            if(err){
                console.log("error occurred",err);
            }
            console.log("file downloaded");
            cleanFolder("uploads");
         })

    }catch(err){
        console.log("error occurred",err);
    }

 })

 app.post('/convertFile/pdf-to-png', uploads.single('file'),
async(req,res,next)=>{
        try{
        if(!req.file){
            res.status(400).send('Please upload a file');
        }
        const pdfFile=req.file.path;
        console.log(pdfFile);
        const outputdir="downloads";
        const outputfile=path.join(outputdir,
`${path.basename(pdfFile,".pdf")}.png`);
        console.log(outputfile);

        const result= await convertapi.convert('png',
{File:pdfFile}, 'pdf');
         await result.saveFiles(outputfile);
```

```
            res.download(outputfile,(err)=>{
                if(err){
                    console.log("error occurred",err);
                }
                console.log("file downloaded");
                cleanFolder("uploads");
            })

        }catch(err){
            console.log("error occurred",err);
        }

    })



app.post("/convertFile/jpg-to-docx", uploads.single("file"),
async (req, res) => {
    try {
        if (!req.file) {
            return res.status(400).json({ error: "No image
uploaded" });
        }

        const imagePath = req.file.path; // Path to the
uploaded image
        console.log("Received image:", imagePath);
        const imageBuffer = fs.readFileSync(imagePath); //
Read image as buffer

        console.log("Image buffer:", imageBuffer);

        // Create a DOCX document with the image
        const doc = new Document({
            sections: [{
                children: [
```

```
                    new Paragraph({
                        children: [
                            new ImageRun({
                                data: imageBuffer,
                                transformation: { width:
500, height: 500 },
                                type:"jpeg",
                            }),
                        ],
                    }),
                ],
            }],
        });
    console.log(req.file);
        // Generate DOCX as a buffer
        const buffer = await Packer.toBuffer(doc);

        // Set response headers for file download
        res.setHeader("Content-Disposition", 'attachment;
filename="output.docx"');
        res.setHeader("Content-Type",
"application/vnd.openxmlformats-
officedocument.wordprocessingml.document");

        // Send DOCX file as response
        res.send(buffer);

        // Clean up uploaded file
        // fs.unlinkSync(imagePath);
        res.on("finish", () => {
            fs.unlinkSync(imagePath);
        });

    } catch (error) {
        console.error("Error:", error);
        res.status(500).json({ error: "Internal server
error" });
    }
});
```

```
app.post('/convertFile/jpg-to-pdf', uploads.single('file'),
async(req,res,next)=>{
    try{
    if(!req.file){
    res.status(400).send('Please upload a file');
    }
    const jpgfile=req.file.path;
    console.log(jpgfile);
    const outputdir="downloads";
    const { name } = path.parse(jpgfile);


    //Another way to extract extension from filename and
work with extension
    // const extension =
path.extname(jpgfile).toLowerCase().replace('.', ''); //
Extract correct extension
    // const result = await convertapi.convert('png', {
File: jpgfile }, extension);


    const outputfile=path.join(outputdir, `${name}.pdf`);
    console.log(outputfile);



      const result= await convertapi.convert('pdf',
{File:jpgfile}, 'jpg');
      console.log(result);
        await result.saveFiles(outputfile);
        res.download(outputfile,(err)=>{
            if(err){
                console.log("error occurred",err);
            }
            console.log("file downloaded");
            cleanFolder("uploads");
        })
    }catch(err){
```

```
        console.log("error occurred",err);
    }
})

app.post('/convertFile/jpg-to-png', uploads.single('file'),
async(req,res,next)=>{


    try{
    if(!req.file){
        res.status(400).send('Please upload a file');
    }

    const jpgfile=req.file.path;
    console.log(jpgfile);
    const outputdir="downloads";

    const { name } = path.parse(jpgfile);  //desructuring
the 'path' object
    // const outputfile=path.join(outputdir,
`${path.basename(jpgfile,".jpg")}.png`);  //work only for
lowercase extension
    const outputfile = path.join(outputdir,
`${name}.png`);  //work for both lowercase and uppercase
extension

    console.log(outputfile);
    const result= await convertapi.convert('png',
{File:jpgfile}, 'jpg');
    await result.saveFiles(outputfile);
    res.download(outputfile,(err)=>{
        if(err){
            console.log("error occurred",err);
        }
        console.log("file downloaded");
        cleanFolder("uploads");
    })
}catch(err){
    console.log("error occurred",err);
```

```
}
});



app.post("/convertFile/png-to-docx", uploads.single("file"),
async (req, res) => {
    try {
        if (!req.file) {
            return res.status(400).json({ message: "No file
uploaded" });
        }

        const pngPath = req.file.path; // Path to uploaded
PNG
        const outputDir = path.join(__dirname, "converted");

        if (!fs.existsSync(outputDir)) {
            fs.mkdirSync(outputDir);
        }
        //   const { name } = path.parse(pngPath);  //
finding name using destructuring the 'path' object
         const name=path.parse(pngPath).name;  // finding
name by extracting name from path object
        const outputDocxPath = path.join(outputDir,
`${name}.docx`);


        // Read the image into a buffer
        const imageBuffer = fs.readFileSync(pngPath);

        // Create a new Aspose.Words Document
        const doc = new aw.Document();
        const builder = new aw.DocumentBuilder(doc);

        // 🔹 Fix: Insert image **directly** from buffer (No
Base64 conversion needed)
        builder.insertImage(imageBuffer);
```

```
        // Save the document as DOCX
        await doc.save(outputDocxPath);

        // Send the converted file as a response
        res.download(outputDocxPath, "converted.docx", (err)
=> {
            if (err) console.error("Error sending file:",
err);
        });

        // Cleanup: Delete files **after response is sent**
        res.on("finish",   () => {
             fs.unlinkSync(pngPath);
             fs.unlinkSync(outputDocxPath);
            console.log("Temporary files deleted.");
        });


        // await cleanupFiles([pngPath, outputDocxPath]);
//  Another way Cleanup temp files

    } catch (error) {
        console.error("Error converting PNG to DOCX:",
error);
        res.status(500).json({ error: "Internal Server
Error" });
    }
});


app.post('/convertFile/png-to-pdf', uploads.single('file'),
async(req,res,next)=>{
    try{
    if(!req.file){
        res.status(400).send('Please upload a file');
    }
    const pngfile=req.file.path;
    console.log(pngfile);
```

```
    const outputdir="downloads";
    const outputfile=path.join(outputdir,
`${path.basename(pngfile,".png")}.pdf`);
    console.log(outputfile);

    const result= await convertapi.convert('pdf',
{File:pngfile}, 'png');
     await result.saveFiles(outputfile);
    // await result.saveFiles(outputfile);

     res.download(outputfile,(err)=>{
        if(err){
            console.log("error occurred",err);
        }
        console.log("file downloaded");
        cleanFolder("uploads");
     })

}catch(err){
    console.log("error occurred",err);
}
})

app.post('/convertFile/png-to-jpg', uploads.single('file'),
async(req,res,next)=>{
    try{
    if(!req.file){
        res.status(400).send('Please upload a file');
    }
    const pngfile=req.file.path;
    console.log(pngfile);
    const outputdir="downloads";
    const outputfile=path.join(outputdir,
`${path.basename(pngfile,".png")}.jpg`);
    console.log(outputfile);

    const result= await convertapi.convert('jpg',
{File:pngfile}, 'png');
     await result.saveFiles(outputfile);
```

```
    // await result.saveFiles(outputfile);

     res.download(outputfile,(err)=>{
        if(err){
            console.log("error occurred",err);
        }
        console.log("file downloaded");
        cleanFolder("uploads");
        fs.unlink(outputfile, (unlinkErr) => {
            if (unlinkErr) {
                console.error("Error deleting file:",
unlinkErr);
            } else {
                // console.log("File deleted after
download");
            }
     })

});
}catch(err){
    console.log("error occurred",err);
}
})


 app.listen(3001,()=>{
 console.log(`Server is  listening on port ${port}`);
});
```

Utils folder

CleanUploadFolder.js

This file is used to make the repeatation of a

Method Or steps wrapped in a function

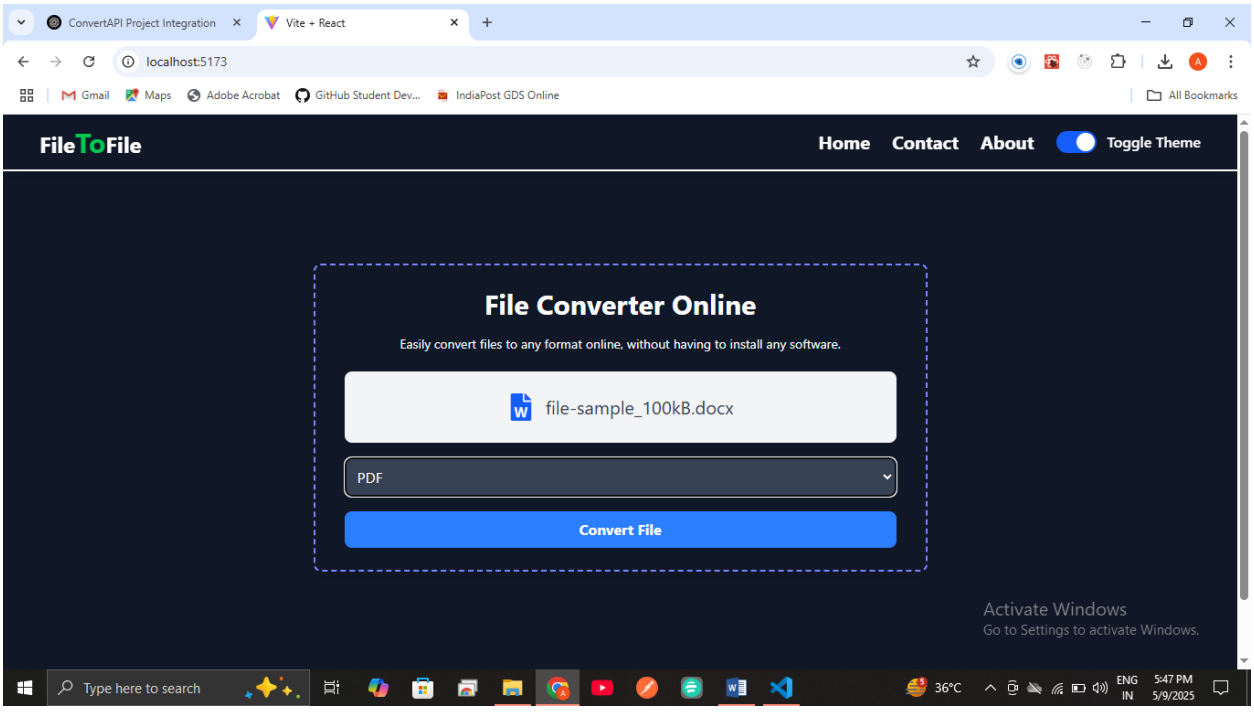
```
const fs=require('fs');
const path=require('path');

const cleanFolder = (folderPath) => {
    fs.readdir(folderPath, (err, files) => {
      if (err) return console.error("Error reading folder:",
err);
      files.forEach(file => {
        fs.unlink(path.join(folderPath, file), err => {
          if (err) console.error("Error deleting file:",
err);
        });
      });
    });
  }


  module.exports = cleanFolder;
```

ConvertAPI Service

ConvertAPI has been instrumental in streamlining the file conversion process in this project. It enabled seamless and reliable transformations between various file formats such as DOCX, PDF, JPG, and PNG, eliminating the need to handle complex conversion logic manually. By integrating ConvertAPI into the backend, the project achieved fast and accurate conversions with minimal setup, saving significant development time. Its comprehensive documentation and robust API endpoints allowed smooth handling of both document and image formats, making it a key component in delivering a user-friendly and efficient file converter application.

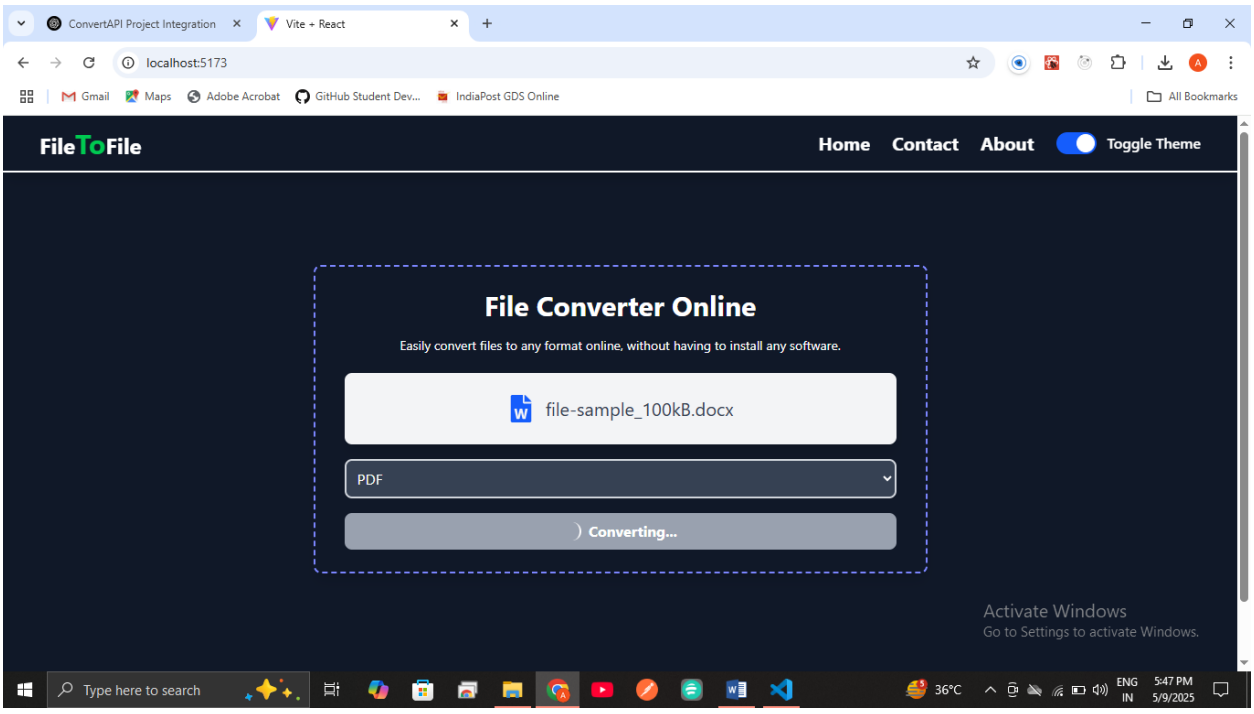
How ConvertAPI Works (Simplified Diagram Explanation)

1. **User Uploads a File**
    - On your frontend (e.g., React), a user selects and uploads a file (DOCX, PDF, PNG, etc.).
    - The file is sent to your backend (e.g., using Express and Multer).



Backend Calls ConvertAPI

- Your backend (Node.js) receives the file and sends a **POST request** to the
- ConvertAPI endpoint.
- You specify:
  - **Input format** (e.g., `docx`)
  - **Target format** (e.g., `pdf`)
  - Your **ConvertAPI secret key**
  - The **file itself**, as form data



2. **ConvertAPI Processes the Conversion**
    - ConvertAPI takes your input file.
    - It processes and converts it on its cloud servers into the desired output format.
    - It then provides a URL or stream to download the converted file.
3. **Backend Sends File Back**
    - Your backend retrieves the converted file (either by downloading it or passing the URL to the frontend).
    - It sends the converted file back to the frontend or triggers an automatic download.
4. **User Gets Converted File**
    - The user sees a preview (if enabled) and downloads the new file in the selected format.

Future Scope

Future Scopes of the File Converter Project

1. 🔄 **Support for More File Formats**
   Extend support to additional formats like XLSX, PPTX, HTML, TXT, EPUB, etc., making the tool a one-stop solution for all file conversions.
2. 🧠 **Smart Format Detection**
   Use AI or heuristics to automatically detect input file types and recommend best output formats, improving user experience.
3. 📁 **Batch Conversion**
   Allow users to upload and convert multiple files at once with zip downloads for bulk processing.
4. ☁️ **Cloud Storage Integration**
   Integrate with Google Drive, Dropbox, or OneDrive for uploading and saving converted files directly to/from the cloud.
5. 👤 **User Authentication & Dashboard**
   Add signup/login features and a dashboard to track previous conversions, manage history, and store files temporarily.
6. 📊 **Conversion Analytics**
   Track file types converted, time saved, user stats, and display visual analytics using charts.
7. ⚙️ **Drag-and-Drop UI & File Preview**
   Make the interface more interactive with drag-and-drop

functionality and live preview of files before and after conversion.

8. 🔐 **Premium Features with Subscriptions**
   Offer a freemium model with paid plans for higher limits, priority conversions, larger file sizes, or premium formats.
9. 💻 **Desktop or Mobile App**
   Build a cross-platform desktop app using Electron or a mobile app using React Native for offline conversions (via local libraries or hybrid models).
10. 📤 **Email or Link Sharing**
    Let users send converted files via email or generate shareable download links with optional expiration.

Conclusion

In conclusion, this file converter web application successfully provides users with an efficient and user-friendly platform for converting files between multiple formats including DOCX, PDF, JPG, and PNG. By integrating React on the frontend and Node.js with Express on the backend, the application ensures fast and seamless file handling. CloudConvert and ConvertAPI were utilized for reliable and accurate file format conversions. The project also implemented secure file upload and download functionality, offering a complete and responsive experience across devices. Overall, this project demonstrates a practical application of full-stack development skills, API integration, and modern web technologies to solve real-world file compatibility issues.

References

1. CloudConvert API Documentation. *CloudConvert*. Retrieved from https://cloudconvert.com/api
2. ConvertAPI Documentation. *ConvertAPI*. Retrieved from https://www.convertapi.com/doc
3. Node.js Documentation. *Node.js Foundation*. Retrieved from https://nodejs.org/en/docs
4. Express.js Guide. *Express*. Retrieved from https://expressjs.com/
5. React Documentation. *Meta (Facebook)*. Retrieved from https://react.dev/
6. Multer Middleware Documentation. *Multer*. Retrieved from https://github.com/expressjs/multer
7. Tailwind CSS Documentation. *Tailwind Labs*. Retrieved from https://tailwindcss.com/docs
8. Axios GitHub Repository. *Axios*. Retrieved from https://github.com/axios/axios

---

Bibliography

- E. Freeman, E. Robson, *Head First HTML and CSS*, O'Reilly Media, 2012.
- B. Eich, *JavaScript: The Definitive Guide*, O'Reilly Media, 2020.
- M. Meyer, *Node.js in Action*, Manning Publications, 2019.

- A. Banks and E. Porcello, *Learning React*, O'Reilly Media, 2020.
- Mozilla Developer Network (MDN). *Web Development Documentation*. Retrieved from https://developer.mozilla.org/