**✅ What Has Been Done in the React UI**

1. **File Upload (Single File):**

* Users can upload a file via drag-and-drop or file picker.
* Only one file at a time is supported.

1. **Format Selection:**

* Users select the source format (auto-detected from file extension if possible).
* Users select the destination format from a list of supported conversions.

1. **Conversion Initiation:**

* A “Convert File” button triggers the conversion process.

1. **Progress & Status Feedback:**

* Visual feedback for uploading, converting, and completion.
* Error messages for unsupported/corrupted files or failed conversions.

1. **Download Link:**

* After conversion, a download link/button is provided for the converted file.

1. **Responsive Design & UX:**

* The UI is responsive and user-friendly.
* Ad placeholders are present for future monetization.

1. **API Integration Points:**

* The UI expects to interact with a backend via HTTP API endpoints for:
* Uploading/converting files
* Checking conversion status
* Downloading converted files
* Fetching supported formats

**🛠️ What Needs to Be Done in the Python Backend**

The backend must provide a REST API that matches the UI’s expectations. The following endpoints and behaviors are required:

**1. POST /convert**

* **Request:** multipart/form-data with fields:
* file: the uploaded file
* sourceFormat: string (e.g., "PDF")
* destinationFormat: string (e.g., "DOCX")
* **Response:** JSON with a unique jobId or direct download info (if synchronous).
* **Behavior:** Start the conversion process, store the job status, and return a job identifier.

**2. GET /status/<jobId>**

* **Request:** jobId as a URL parameter.
* **Response:** JSON with status info:
* status: one of pending, converting, completed, error
* progress: percentage (optional)
* downloadUrl: present if completed
* error: error message if failed

**3. GET /download/<jobId>**

* **Request:** jobId as a URL parameter.
* **Response:** The converted file as a downloadable attachment.

**4. GET /formats**

* **Response:** JSON array of supported conversion formats, matching the structure used in the UI (see the roadmap or UI code for the format).

**5. Error Handling**

* Return clear error messages for unsupported formats, file validation failures, or conversion errors.

**6. File Handling**

* Securely handle file uploads and temporary storage.
* Clean up temporary files after a set period or after download.

**7. Conversion Logic**

* Use appropriate Python libraries for file conversion (e.g., PyPDF2, python-docx, Pillow, ffmpeg, etc.).
* Support all source/destination format pairs listed in the UI’s format JSON.

**8. CORS**

* Enable CORS so the React frontend can communicate with the backend.

**📝 Example API Contract**

**POST /convert**

**http**

**Apply to Todo.docx**

POST /convert

Content-Type: multipart/form-data

file: <uploaded file>

sourceFormat: PDF

destinationFormat: DOCX

**Response:**

json

Apply to Todo.docx

{

  "jobId": "abc123"

}

**GET /status/abc123**

**json**

**Apply to Todo.docx**

{

  "status": "completed",

  "progress": 100,

  "downloadUrl": "/download/abc123"

}

**GET /download/abc123**

* Returns the converted file as an attachment.

**GET /formats**

**json**

**Apply to Todo.docx**

[

  {

    "source": "PDF",

    "destination": ["DOCX", "DOC", *...*]

  },

*...*

]

**🟢 Summary Table**

| **UI Feature** | **Backend Endpoint Needed** | **Description** |
| --- | --- | --- |
| File upload & convert | POST /convert | Accept file, start conversion, return jobId |
| Conversion status | GET /status/<jobId> | Return status, progress, downloadUrl |
| Download converted file | GET /download/<jobId> | Serve converted file |
| Format selection | GET /formats | List supported conversions |
| Error handling | All endpoints | Return clear error messages |

**If you implement these endpoints and behaviors in Python (using Flask, FastAPI, or Django), your backend will be fully compatible with the current React UI.**Let me know if you want a Python starter template for this backend!