```
</html>
```

It looks like you're building a web application that allows users to upload an Excel file, display its contents, and then search within the data. Are you looking for feedback, improvements, or additional functionality for this project? Let me know how I can assist!

give the details for this CHOICE OF COMPONENTS / MODULES / METHODS/TECHNIQUES EQUIPMENT USED FOR PROJECT DEVELOPMENT

Here's a breakdown of the **Choice of Components, Modules, Methods/Techniques, and Equipment Used for Project Development** based on your provided HTML and JavaScript code:

---

1. Components Used

Frontend

- **HTML5** – Used for structuring the web pages.
- **CSS3** – Used for styling the interface.
- **JavaScript** – Used for client-side scripting and handling user interactions.

Backend (Assumed)

- **Django (Python-based framework)** – The use of `{% include 'Main.html' %}` and `{% load static %}` suggests Django for rendering templates and serving static files.

Libraries & Frameworks

- **XLSX.js** – A JavaScript library (`xlsx.full.min.js`) used to parse and display Excel files.
- **Boxicons** – Used for adding icons to the search input field.

---

2. Modules Used

- **File Handling Module** – Handles Excel file uploads and parsing.
- **LocalStorage Module** – Stores Excel data in the browser for retrieval between pages.
- **Search Module** – Provides an auto-suggestion feature and filters Excel data dynamically.
- **Profile Display Module** – Dynamically loads user profile images based on search results.

---

3. Methods/Techniques Used

Client-Side Processing

- **FileReader API** – Reads Excel files as an `ArrayBuffer` for processing.
- **XLSX.js Utilities** – Converts Excel sheets to HTML tables (`XLSX.utils.sheet_to_html`).

- **LocalStorage** – Used to persist table data across pages without requiring a database.

Search Algorithm

- **String Matching (Case-Insensitive Search)** – Searches within table cells by comparing input text with content from the Excel sheet.
- **Auto-Suggestions** – Extracts and displays possible matches dynamically from Excel data.

Dynamic UI Updates

- **JavaScript DOM Manipulation** – Updates search results and displays tables dynamically.
- **Event Listeners** – Listens for file input changes and search input updates.

Error Handling

- **Image Fallback Mechanism** – Tries loading `` `.png` `` images first, then `` `.jpg` `` as a fallback.
- **Empty Search Handling** – Displays appropriate messages when no results are found.

---

4. Equipment Used for Development

Software Tools

- **Code Editor (VS Code, PyCharm, Sublime Text, etc.)** – For writing and debugging code.
- **Web Browser (Chrome, Firefox, Edge, etc.)** – For testing the web application.
- **Django Framework** (Assumed) – For backend integration.
- **Excel Software (MS Excel, Google Sheets, etc.)** – For preparing sample data.

Hosting & Deployment (If Applicable)

- **Localhost (During Development)** – Using Django's built-in server (`` `python manage.py runserver` ``).
- **Cloud Hosting (If Deployed)** – Platforms like AWS, Heroku, or PythonAnywhere.

This summary provides a structured overview of the **components, modules, methods, and equipment** used in your project. Let me know if you need more details! 🚀