**1. Overview**

**1.1 Purpose**

The purpose of this document is to define the software requirements for a File Converter Website. This website will enable users to upload files in Word and convert them into PDF format and vice-versa. The Frontend will be developed using HTML and CSS, while the back end will be implemented in JavaScript.

**1.2 Scope**

The File Converter Website aims to provide an easy-to-use interface for users to convert files between Word to PDF formats. The system will support file uploads, format selection, conversion, and downloading of the converted file.

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* **HTML**: Hyper Text Markup Language
* **CSS**: Cascading Style Sheets
* **API**: Application Programming Interface

**1.4 Overall Description**

**2.Product Perspective**

The File Converter Website will be a standalone web application with no dependencies on external systems except for standard libraries and APIs for file conversion.

**2.1Product Features**

* **User-Friendly Interface**: Simple and intuitive interface for uploading and downloading files.
* **File Format Support**: Support for converting documents (e.g., PDF to DOCX),
* **Real-Time Conversion**: Immediate file conversion upon user request.
* **Error Handling**: Alerts for unsupported file formats or conversion failures.

**2.1 User Classes and Characteristics**

* **General Users**: Individuals who want to convert files without requiring technical expertise.

**2.2 Operating Environment**

* **Frontend**: HTML and CSS
* **Backend**: Java script

**2.3 Assumptions and Dependencies**

* Files uploaded by users will not exceed the size limit defined by the system.
* Supported file formats are predefined and limited.

**2.4 Specific Requirements**

**3. Functional Requirements**

1. **File Upload**: Users can upload files from their local system.
2. **Format Selection**: Users can select the desired output format from a dropdown menu.
3. **File Conversion**: The backend will process and convert the uploaded file to the PDF format.
4. **Download Option**: Users can download the converted file.
5. **Error Notifications**: The system will notify users of unsupported formats or conversion errors.

**3.1 Non-Functional Requirements**

1. **Performance**: The system should process file conversions for files under 10MB.
2. **Scalability**: The system should handle up to 100 simultaneous users.
3. **Security**: Uploaded files should be deleted from the server after conversion.

**3.2 External Interface Requirements**

* **User Interface**: Developed using HTML and CSS for simplicity and responsiveness.
* **Backend Interface**: Java script-based API to handle file uploads, processing, and downloads.

**3.3 Future Enhancements**

* Support for batch file conversion.
* Advanced format options.

**3.4 Glossary**

* **Frontend**: The part of the website visible to the user.
* **Backend**: The server-side logic and operations of the website.

**4. Use case diagram**

User

upload word

Or pdf file

convert to pdf

download pdf

Or word file

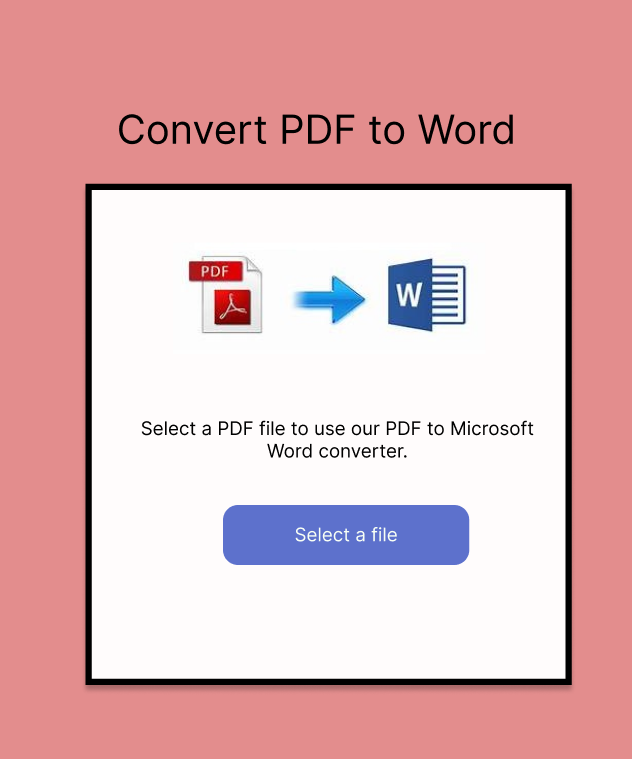
System administrator

manage

system

setting

**4.1 wireframe:**



**4.2 Stack:**

**1.Frontend:**

Purpose: Create a user-friendly interface**.**

* HTML
* CSS
* **Framework:**
* **UI Libraries:**
* **Key Features:**
  + File upload form
  + Conversion progress indicator
  + Download button for converted files

**2.Backend:**

Purpose: Handle file processing, user requests, and secure conversions

* Java Script

**Language:**

* Node.js
* JavaScript

**Framework:**

**Storage:**

* Store uploaded files temporarily using Amazon S3, MYSQL or a local disk**.**

**File Processing**

Purpose: Convert and manage file integrity**.**

**Libraries/Tools:**

**LibreOffice**

**4. Storage & Caching**

Purpose: Handle file storage and speed up conversions**.**

* **Storage:**

**Caching:**

1. **Security**

Purpose: Protect user data and files.

* **Encryption:**
* **Validation:**

Scan files for malware

* **File Deletion:**

**6. Deployment**

Purpose: Host and run your website**.**

**Hosting Platforms:**

**7. SEO and Analytics**

Purpose: Improve discoverability and monitor performance**.**

**SEO Optimization:**

Use meta tags, sitemaps, and schema.org for structured data.

Optimize page speed using Lighthouse and CDN.

**Analytics:**

Google Analytics for traffic insights**.**