**Human Computer Interaction HCI : Theory ISE 1 of 2**

**Development Activity**

1. Develop the User Interface for the following task by applying HCI Screen design and Layout guidelines : Schneiderman’s eight golden rules.

2. Use React platform with different state of the art UI frameworks. No backend database / No server side coding

**Activity - 1 :** Menu-driven word-processing system

Group the following functions under appropriate headings, assuming that they are to form the basis. The headings you choose will become the menu titles, with the functions appearing under the appropriate one. You can choose as many or as few menu headings as you wish. You may also alter the wordings of the functions slightly if you wish.

*save, save as, new, delete, open mail, send mail, quit, undo, table, glossary, preferences, character style, format paragraph, lay out document, position on page, plain text, bold text, italic text, underline, open file, close file, open copy of file, increase point size, decrease point size, change font, add footnote, cut, copy, paste, clear, repaginate, add page break, insert graphic, insert index entry, print, print preview, page setup, view page, find word, change word, go to, go back, check spelling, view index, see table of contents, count words, renumber pages, repeat edit, show alternative document, help.*

Grouping of the functions under appropriate headings.

### **File**

* New
* Open File
* Open Copy of File
* Save
* Save As
* Close File
* Delete
* Quit

### **Edit**

* Undo
* Cut
* Copy
* Paste
* Clear
* Find Word
* Change Word
* Go To
* Go Back
* Repeat Edit
* Check Spelling
* Count Words

### **View**

* Print Preview
* Page Setup
* View Page
* Show Alternative Document
* View Index
* See Table of Contents
* Glossary

### **Format**

* Character Style
* Plain Text
* Bold Text
* Italic Text
* Underline
* Increase Point Size
* Decrease Point Size
* Change Font
* Format Paragraph
* Lay Out Document
* Position on Page

### **Insert**

* Add Footnote
* Add Page Break
* Insert Graphic
* Insert Index Entry
* Table

### **Tools**

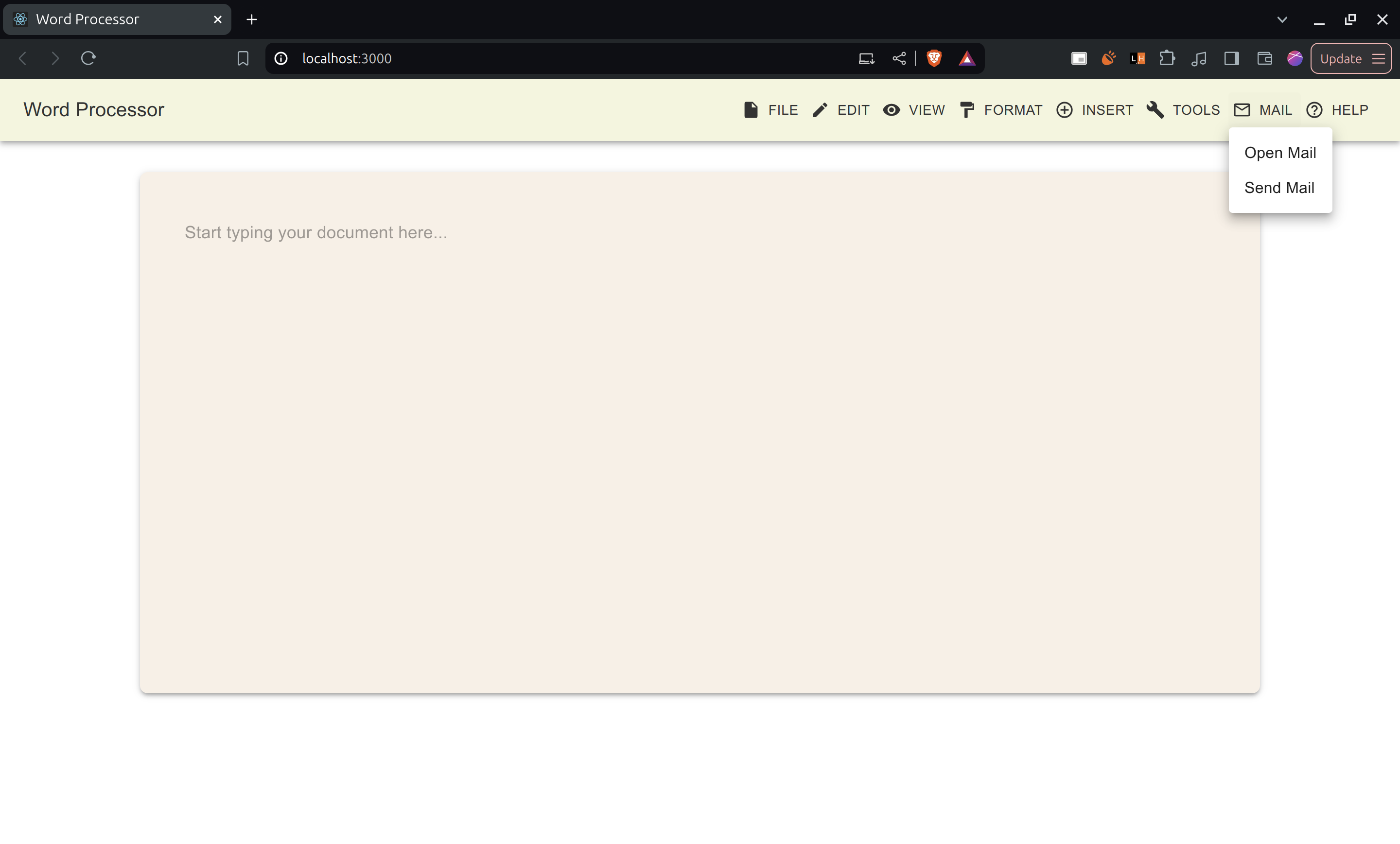
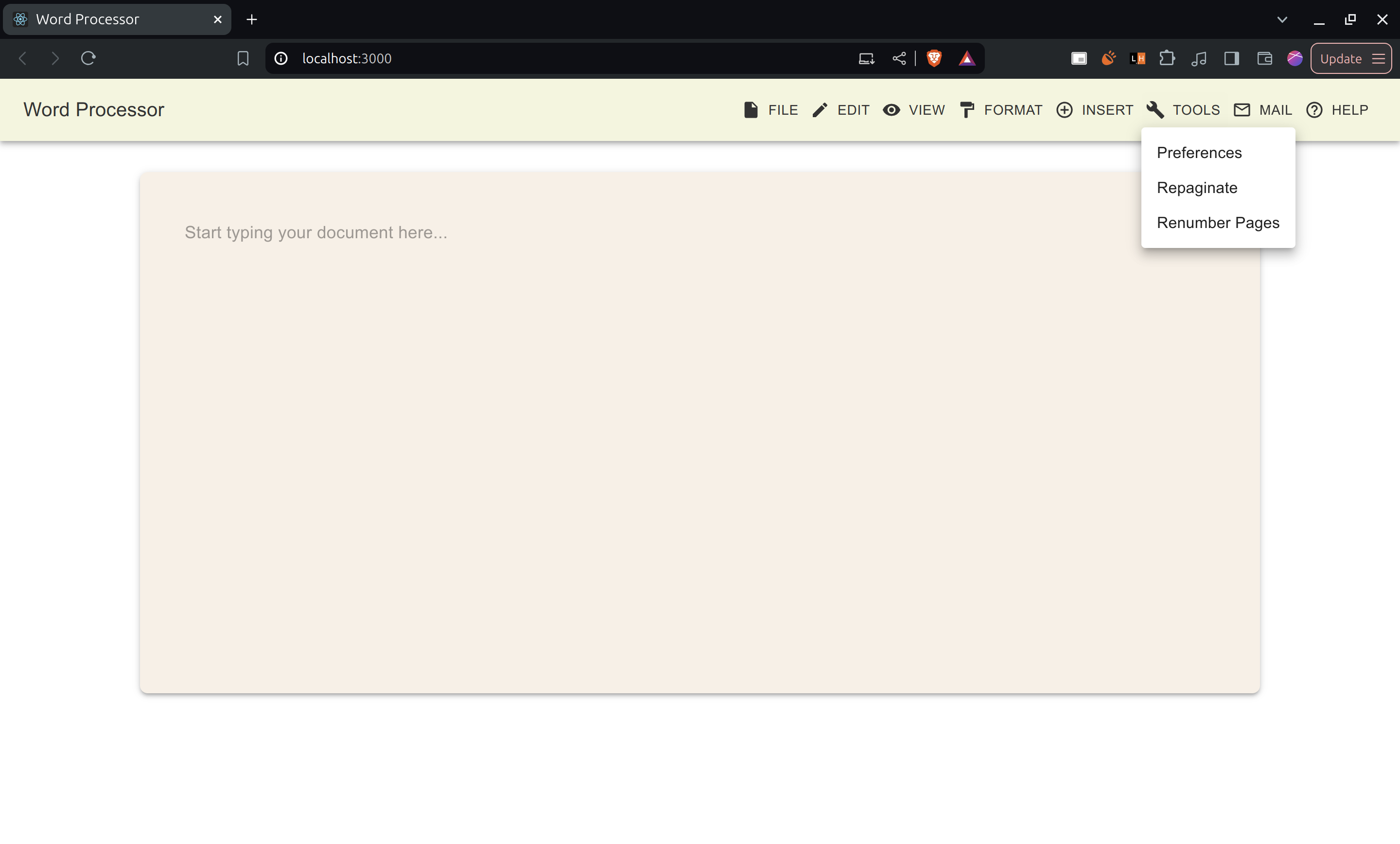
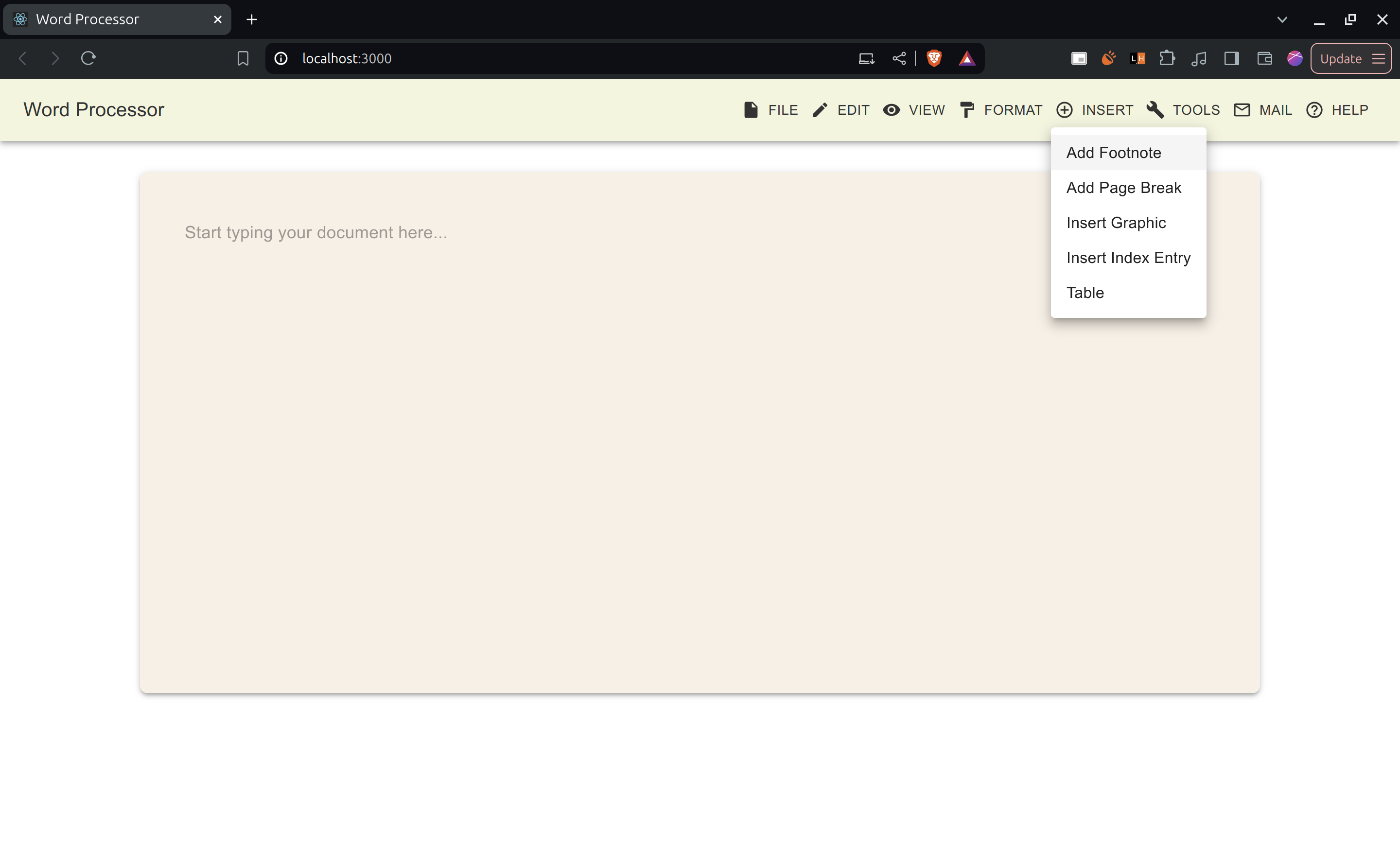
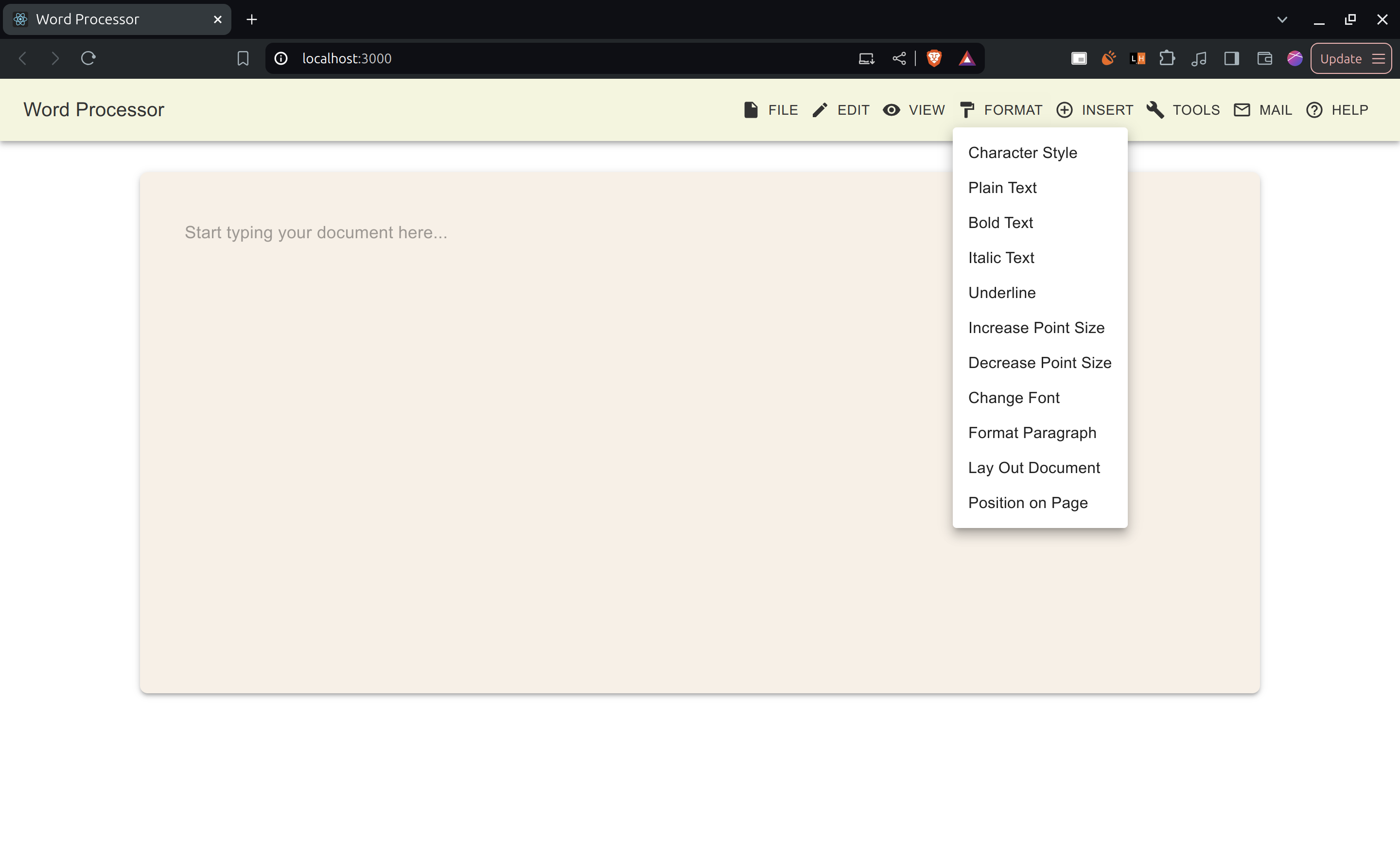
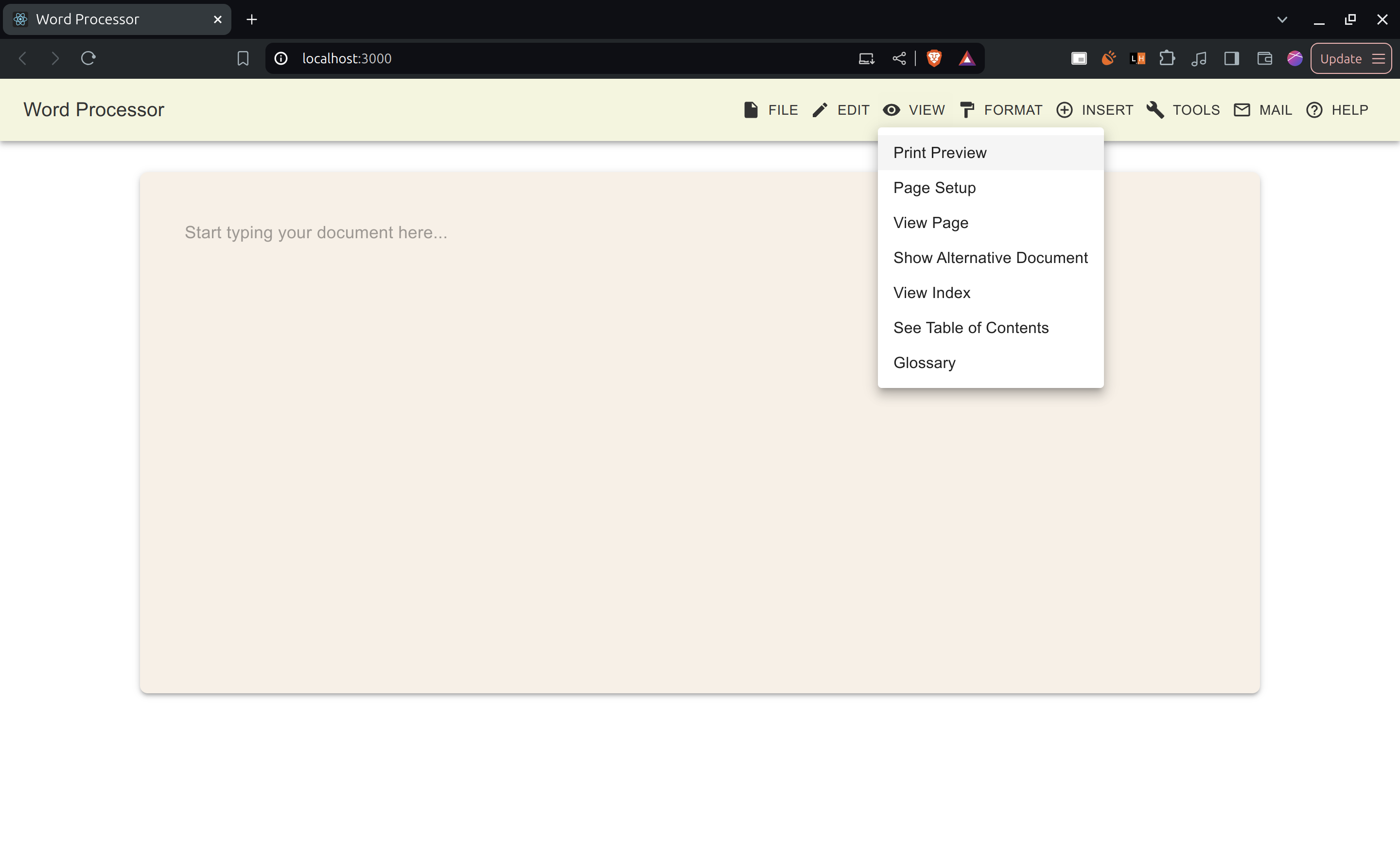
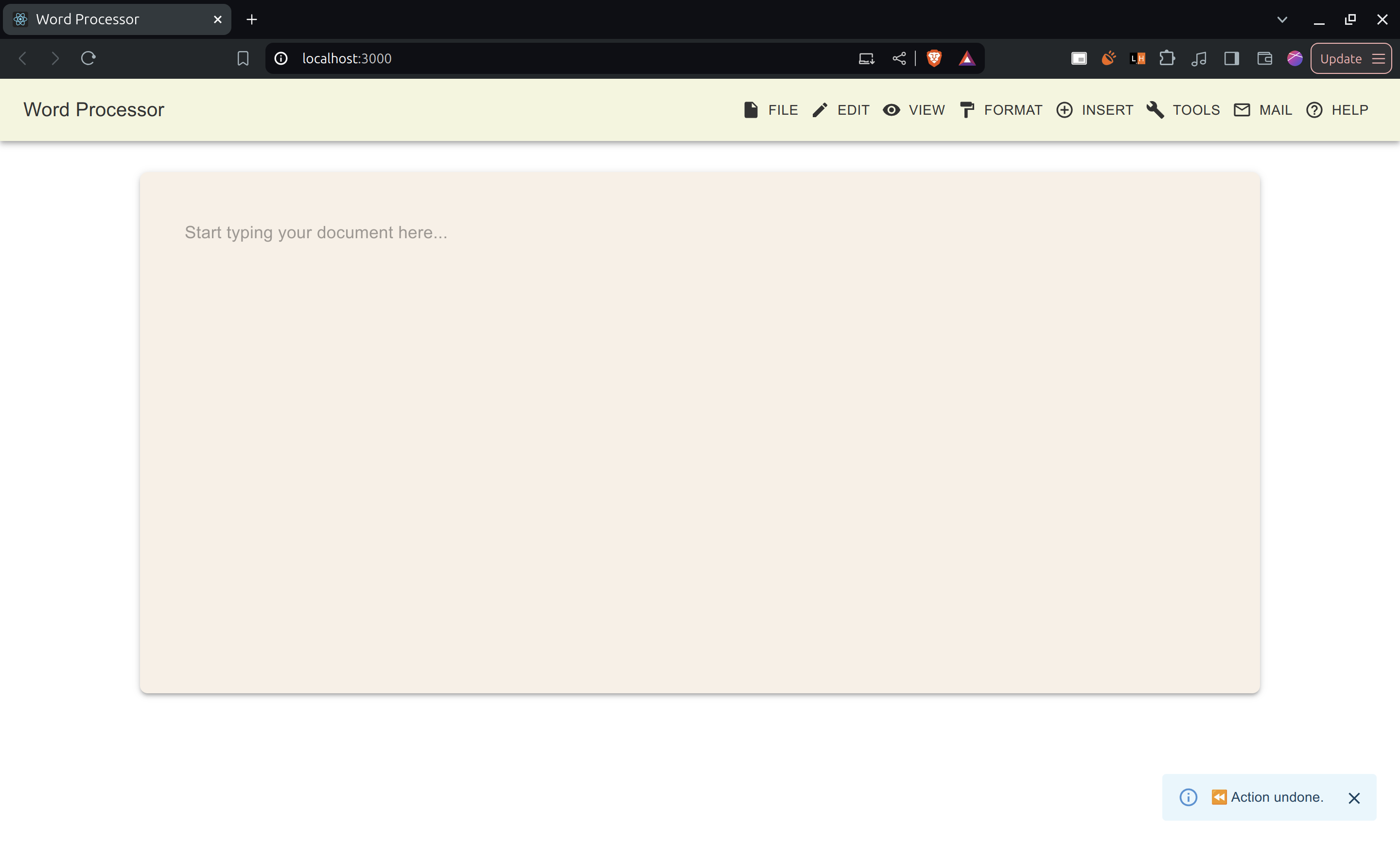
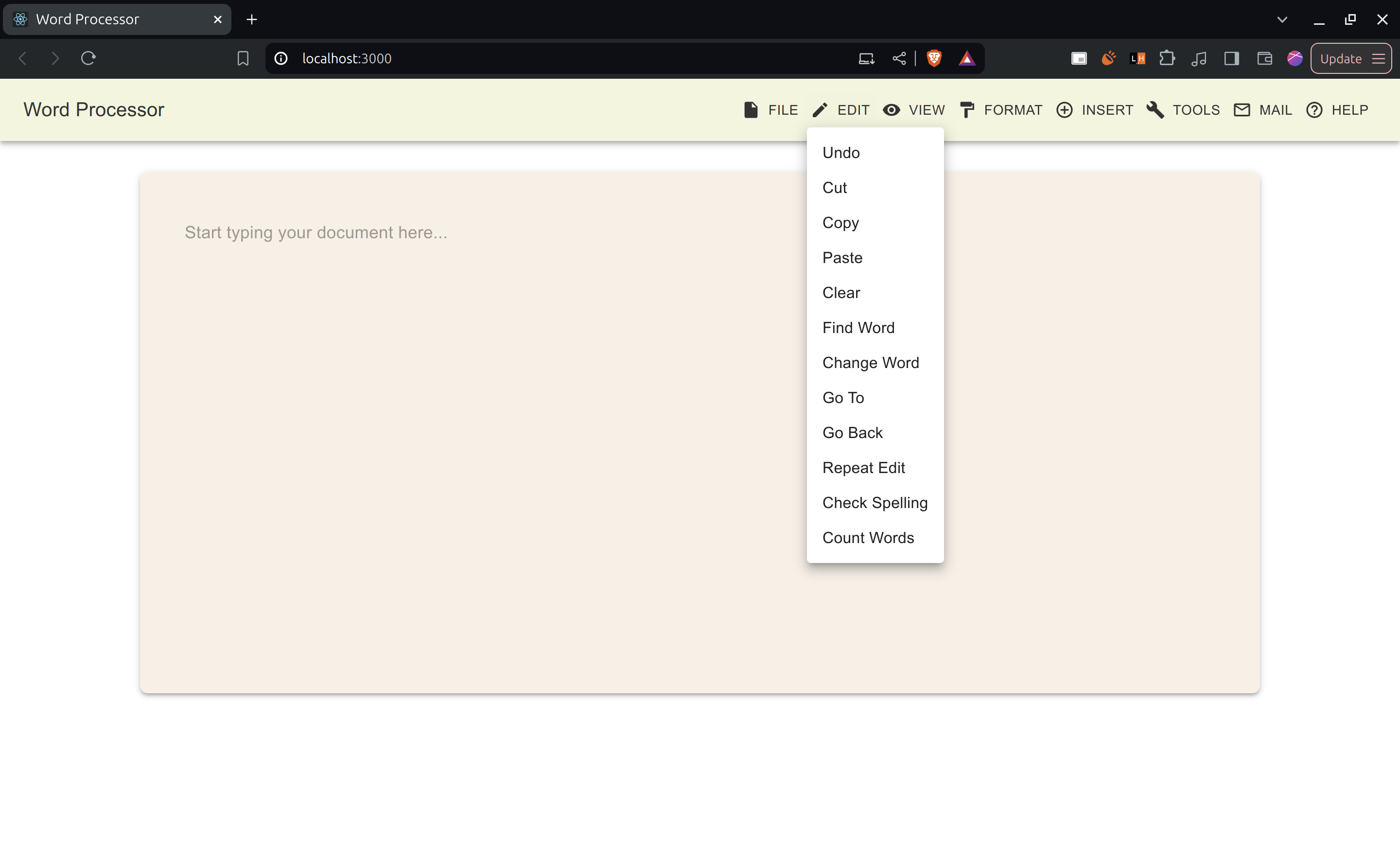
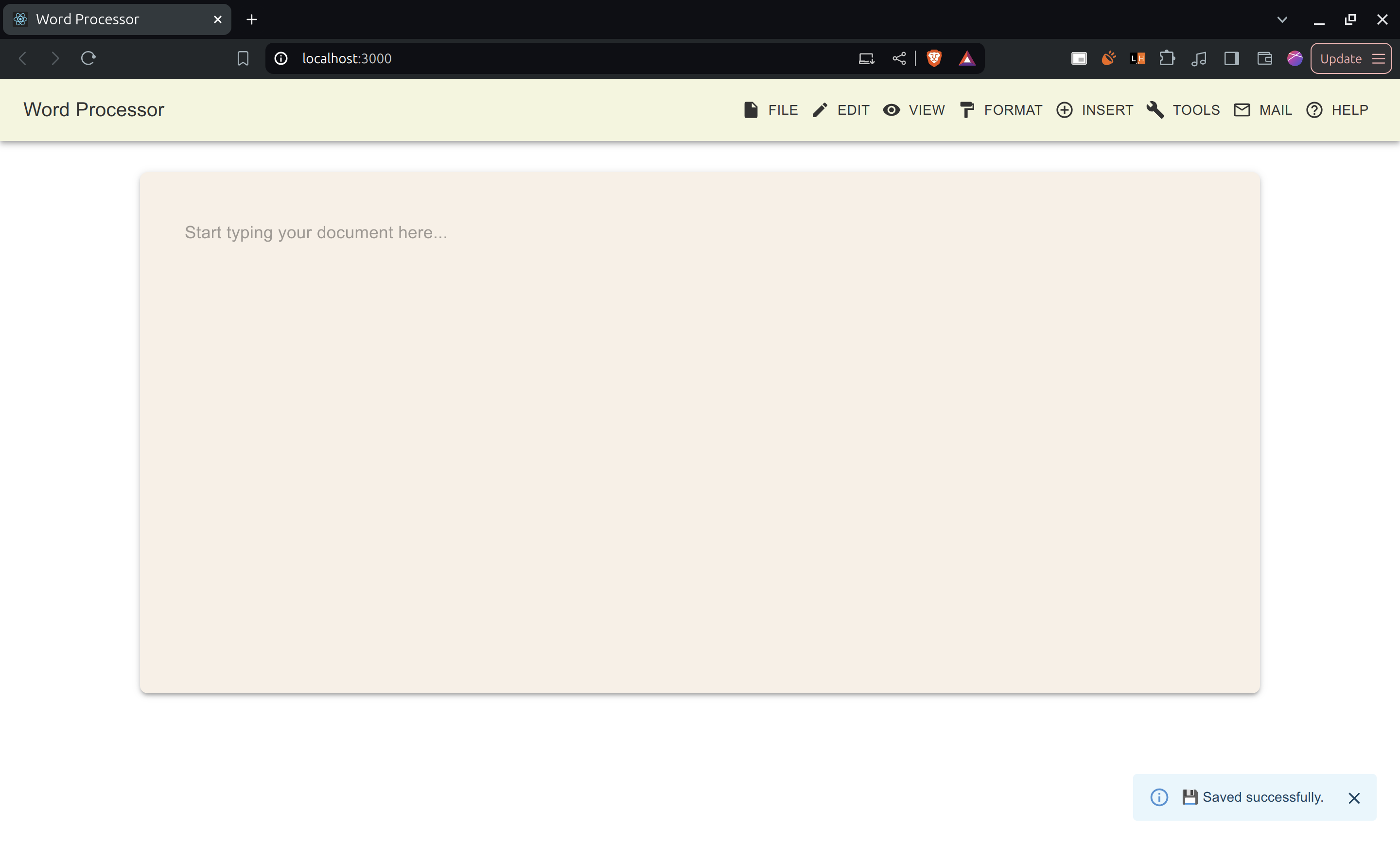
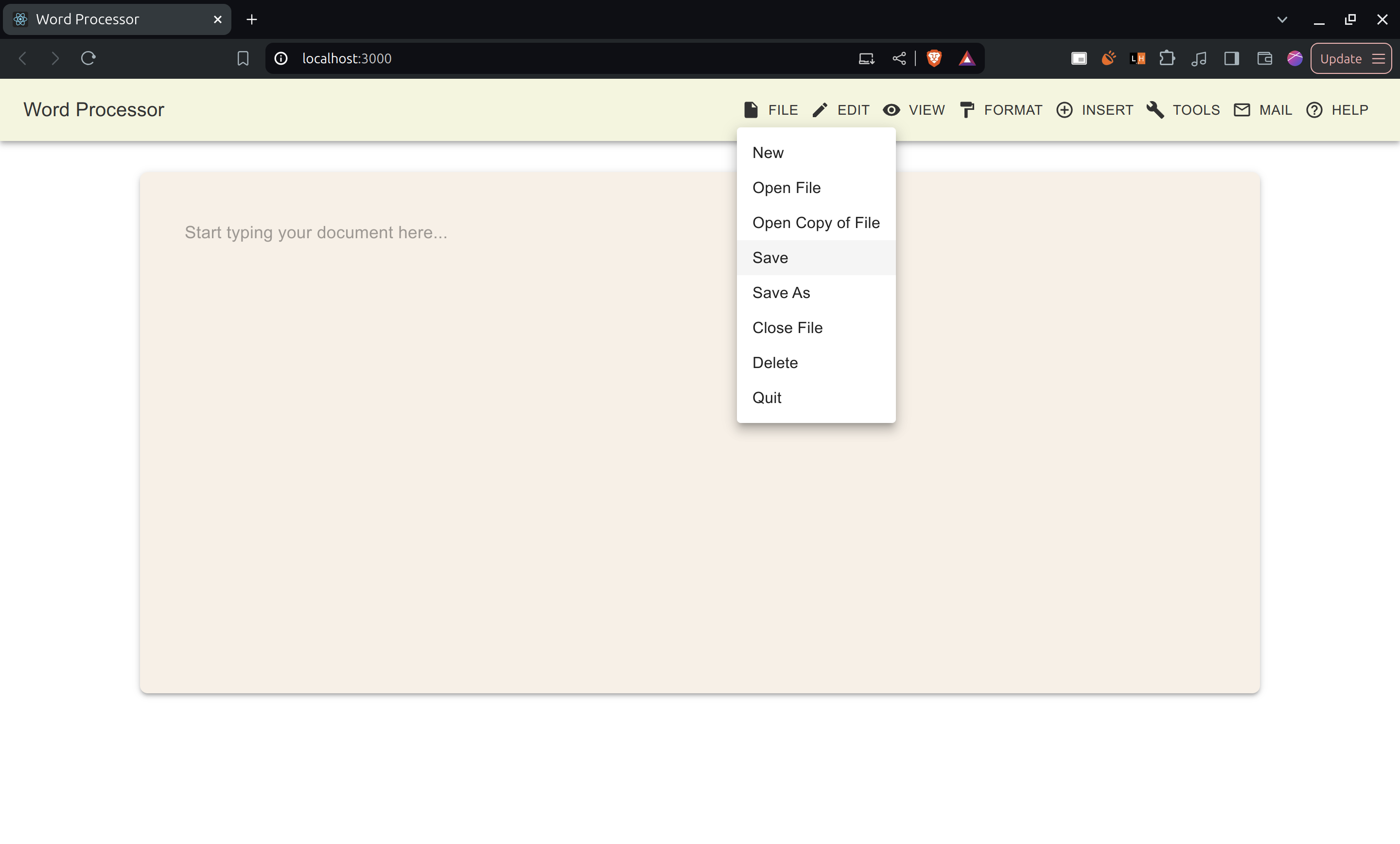
* Preferences
* Repaginate
* Renumber Pages

### **Mail**

* Open Mail
* Send Mail

### **Help**

* Help



**React UI components using MUI:** [**https://mui.com/**](https://mui.com/)

Snackbar for Toast Notifications:

<Snackbar

open={snackbarOpen}

autoHideDuration={3000}

onClose={handleSnackbarClose}

anchorOrigin={{ vertical: 'bottom', horizontal: 'right' }}

>

<Alert onClose={handleSnackbarClose} severity="info" sx={{ width: '100%' }}>

{snackbarMessage}

</Alert>

</Snackbar>

Menu Buttons with Icons and Labels:

<AppBar position="static" sx={{ backgroundColor: '#F5F5DC', color: '#333' }}>

<Toolbar>

<Typography variant="h6" sx={{ flexGrow: 1, color: '#333' }}>

Word Processor

</Typography>

<Button

color="inherit"

startIcon={<FileIcon />}

onClick={(event) => handleClick(event, 'file')}

>

File

</Button>

{renderMenu('file', ['New', 'Open File', 'Open Copy of File', 'Save', 'Save As', 'Close File', 'Delete', 'Quit'])}

<Button

color="inherit"

startIcon={<EditIcon />}

onClick={(event) => handleClick(event, 'edit')}

>

Edit

</Button>

{renderMenu('edit', ['Undo', 'Cut', 'Copy', 'Paste', 'Clear', 'Find Word', 'Change Word', 'Go To', 'Go Back', 'Repeat Edit', 'Check Spelling', 'Count Words'])}

<Button

color="inherit"

startIcon={<ViewIcon />}

onClick={(event) => handleClick(event, 'view')}

>

View

</Button>

{renderMenu('view', ['Print Preview', 'Page Setup', 'View Page', 'Show Alternative Document', 'View Index', 'See Table of Contents', 'Glossary'])}

<Button

color="inherit"

startIcon={<FormatIcon />}

onClick={(event) => handleClick(event, 'format')}

>

Format

</Button>

{renderMenu('format', ['Character Style', 'Plain Text', 'Bold Text', 'Italic Text', 'Underline', 'Increase Point Size', 'Decrease Point Size', 'Change Font', 'Format Paragraph', 'Lay Out Document', 'Position on Page'])}

<Button

color="inherit"

startIcon={<InsertIcon />}

onClick={(event) => handleClick(event, 'insert')}

>

Insert

</Button>

{renderMenu('insert', ['Add Footnote', 'Add Page Break', 'Insert Graphic', 'Insert Index Entry', 'Table'])}

<Button

color="inherit"

startIcon={<ToolsIcon />}

onClick={(event) => handleClick(event, 'tools')}

>

Tools

</Button>

{renderMenu('tools', ['Preferences', 'Repaginate', 'Renumber Pages'])}

<Button

color="inherit"

startIcon={<MailIcon />}

onClick={(event) => handleClick(event, 'mail')}

>

Mail

</Button>

{renderMenu('mail', ['Open Mail', 'Send Mail'])}

<Button

color="inherit"

startIcon={<HelpIcon />}

onClick={(event) => handleClick(event, 'help')}

>

Help

</Button>

{renderMenu('help', ['Help'])}

</Toolbar>

</AppBar>

Main Content Area:

<Container sx={{ mt: 4 }}>

<Paper

elevation={3}

sx={{

padding: 4,

minHeight: '60vh',

backgroundColor: '#FAF0E6',

borderRadius: 2,

}}

>

<TextField

fullWidth

multiline

placeholder="Start typing your document here..."

variant="outlined"

InputProps={{

style: {

fontSize: '1.1rem',

lineHeight: '1.6',

fontFamily: 'Arial, sans-serif',

},

}}

sx={{

'& .MuiOutlinedInput-root': {

'& fieldset': {

borderColor: 'transparent',

},

'&:hover fieldset': {

borderColor: '#B0B0B0',

},

'&.Mui-focused fieldset': {

borderColor: '#B0B0B0',

},

},

}}

/>

</Paper>

</Container>

Feedback on click:

const handleMenuItemClick = (item) => {

let message = '';

switch (item) {

case 'New':

message = "✨ New document created.";

break;

case 'Open File':

message = "📂 Opening file.";

break;

case 'Save':

message = "💾 Saved successfully.";

break;

case 'Quit':

message = "👋 Quitting the app.";

break;

case 'Undo':

message = "⏪ Action undone.";

break;

case 'Cut':

message = "✂️ Text cut to clipboard.";

break;

case 'Paste':

message = "📋 Pasted successfully.";

break;

case 'Find Word':

message = "🔍 Searching for word...";

break;

case 'Print Preview':

message = "🖨️ Print queued.";

break;

case 'Send Mail':

message = "📧 Your mail is sent.";

break;

default:

message = `${item} success.`;

}

setSnackbarMessage(message);

setSnackbarOpen(true);

handleClose();

};

### **Schneiderman’s Eight Golden Rules**

1. **Strive for Consistency:**
   1. Followed consistent creme and white color palette.
2. **Enable Frequent Users to Use Shortcuts:**
   1. Provided keyboard shortcuts for common actions (e.g., Ctrl + S for Save, Ctrl + Z for Undo).
   2. Include tooltips with shortcuts on hover.
3. **Offer Informative Feedback:**
   1. Displayed confirmation messages when actions are completed (e.g., "Document saved successfully").
4. **Design Dialogs to Yield Closure:**
   1. As backend was not configured, this rule has not been implemented.
   2. We can ensure that actions like "Save As" clearly guide the user through the process, providing a final confirmation.
5. **Offer Simple Error Handling:**
   1. Provided simple error messages (e.g., "File empty, can’t be saved").
6. **Permit Easy Reversal of Actions:**
   1. As backend was not configured, this rule has not been implemented.
   2. Allow users to cancel actions like file uploads or formatting changes.
7. **Support Internal Locus of Control:**
   1. Predictable actions (e.g., clicking "Save" should always save the current document).
   2. Avoid unexpected changes or actions.
8. **Reduce Short-Term Memory Load:**
   1. Simple interface, showing only essential options at any given time.
   2. Used icons with each menu title.