RegisterClientLocalizationsError Fix

Issue Description

The application was showing a JavaScript error in the browser console:

```
RegisterClientLocalizationsError {message: "Cannot read properties of undefined
(reading 'translations')", name: 'RegisterClientLocalizationsError'}
```

Root Cause Analysis

After thorough investigation, this error was determined to be caused by **browser extensions** rather than the application code itself. The error name “RegisterClientLocalizationsError” does not appear anywhere in the codebase or in standard Next.js/React libraries.

Common sources of this error include:
- Translation/localization browser extensions
- Language learning extensions
- Accessibility extensions
- Ad blockers with localization features

Solution Implemented

1. Error Handler Component (`src/app/error-handler.tsx`)

Created a client-side error handler that:
- Intercepts and suppresses known external errors from browser extensions
- Prevents console pollution from third-party extension errors
- Maintains normal error logging for legitimate application errors

2. Global Error Boundary (`src/app/error.tsx`)

Added a Next.js error boundary that:
- Catches any unhandled errors in the application
- Provides a user-friendly error message
- Offers a “Try again” button to recover from errors

3. Favicon Addition

- Added `/public/favicon.svg` to eliminate 404 errors
- Updated metadata in `src/app/layout.tsx` to reference the favicon
- Improves overall application polish

4. Layout Updates (`src/app/layout.tsx`)

- Integrated the ErrorHandler component
- Added proper favicon references
- Ensured error handling is active across all pages

Testing

To verify the fix:
1. Restart the application: `npm start`
2. Open the browser console
3. Navigate through different pages
4. The RegisterClientLocalizationsError should no longer appear

User Recommendations

If users continue to see localization-related errors:
1. Disable browser extensions one by one to identify the culprit
2. Test in an incognito/private window (extensions are typically disabled)
3. Try a different browser to confirm it's extension-related

Technical Details

- The error handler uses `window.addEventListener('error')` and `window.addEventListener('unhandledrejection')` to catch errors
- Console.error is wrapped to filter out known external errors
- The solution is non-invasive and doesn't affect legitimate error reporting

Files Modified

- `src/app/layout.tsx` - Added ErrorHandler and favicon
- `src/app/error-handler.tsx` - New error suppression component
- `src/app/error.tsx` - New error boundary component
- `public/favicon.svg` - New favicon file
- `LOCALIZATION_ERROR_FIX.md` - This documentation

Future Considerations

If the error persists or new external errors appear:
1. Update the error filter patterns in `error-handler.tsx`
2. Consider adding a user-facing notification about browser extension conflicts
3. Monitor error logs for patterns that might indicate real application issues