Python Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Oppia AI Assistant Prototype</title>
    <style>
        :root {
            --primary-color: #009688;
            --primary-dark: #00796b;
            --secondary-color: #3f51b5;
            --secondary-dark: #303f9f;
            --bg-color: #f0f2f5;
            --card-bg: #ffffff;
            --text-main: #333333;
            --text-secondary: #666666;
            --border-color: #e0e0e0;
            --success-color: #4CAF50;
            --warning-color: #FF9800;
            --error-color: #F44336;
        }

        body {
            font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Rob
oto, Helvetica, Arial, sans-serif;
            background-color: var(--bg-color);
            color: var(--text-main);
            margin: 0;
            padding: 20px;
            line-height: 1.6;
        }

        /* Layout Structure */
        .main-wrapper {
            max-width: 900px;
            margin: 0 auto;
        }

        header {
            text-align: center;
            margin-bottom: 30px;
        }

        header h2 {
            color: var(--primary-color);
            margin: 0;
            font-weight: 700;
        }

```

```
        header p {
            color: var(--text-secondary);
            margin-top: 5px;
        }

        .container {
            display: flex;
            flex-direction: row;
            gap: 24px;
            align-items: flex-start;
        }

        /* Responsive Stack */
        @media (max-width: 768px) {
            .container {
                flex-direction: column;
            }
        }

        /* Panels */
        .editor-pane, .ai-pane {
            flex: 1;
            background-color: var(--card-bg);
            border: 1px solid var(--border-color);
            border-radius: 12px;
            box-shadow: 0 4px 12px rgba(0,0,0,0.05);
            display: flex;
            flex-direction: column;
            overflow: hidden;
        }

        .pane-header {
            padding: 15px 20px;
            border-bottom: 1px solid var(--border-color);
            background-color: #fafafa;
        }

        .pane-header h3 {
            margin: 0;
            font-size: 1.1rem;
            color: var(--text-main);
            display: flex;
            align-items: center;
            gap: 8px;
        }

        .pane-body {
            padding: 20px;
            flex: 1;
        }

        /* Input Area */
```

```
        textarea {
            width: 100%;
            min-height: 150px;
            padding: 12px;
            border: 2px solid var(--border-color);
            border-radius: 8px;
            font-family: inherit;
            font-size: 1rem;
            resize: vertical;
            transition: border-color 0.2s;
            box-sizing: border-box;
            margin-bottom: 15px;
        }

        textarea:focus {
            outline: none;
            border-color: var(--primary-color);
        }

        /* Controls */
        .controls {
            display: flex;
            gap: 10px;
            flex-wrap: wrap;
            margin-bottom: 20px;
        }

        button {
            padding: 10px 18px;
            border: none;
            border-radius: 6px;
            cursor: pointer;
            font-weight: 600;
            font-size: 0.95rem;
            color: white;
            transition: transform 0.1s, opacity 0.2s;
            display: inline-flex;
            align-items: center;
            gap: 6px;
        }

        button:active { transform: scale(0.98); }
        button:disabled { opacity: 0.6; cursor: not-allowed; }

        #btnGen { background-color: var(--primary-color); }
        #btnGen:hover:not(:disabled) { background-color: var(--primary-dar
k); }

        #btnCheck { background-color: var(--secondary-color); }
        #btnCheck:hover:not(:disabled) { background-color: var(--secondary-
dark); }
```

```
        /* AI Output Styles */
        .output-group { margin-bottom: 20px; }
        .output-group h4 {
            margin-bottom: 10px;
            color: var(--text-secondary);
            font-size: 0.9rem;
            text-transform: uppercase;
            letter-spacing: 0.5px;
        }

        .suggestion-item {
            background: #f8f9fa;
            padding: 12px;
            border: 1px solid var(--border-color);
            margin-bottom: 8px;
            border-radius: 6px;
            cursor: pointer;
            transition: all 0.2s;
            position: relative;
        }

        .suggestion-item:hover {
            background-color: #e0f2f1;
            border-color: var(--primary-color);
            transform: translateX(4px);
        }

        .suggestion-item::after {
            content: '+';
            position: absolute;
            right: 12px;
            top: 50%;
            transform: translateY(-50%);
            color: var(--primary-color);
            font-weight: bold;
        }

        /* Quality Badge Styles */
        .quality-badge {
            font-weight: bold;
            padding: 6px 12px;
            border-radius: 20px;
            display: inline-block;
            font-size: 0.85rem;
        }
        .good { color: white; background-color: var(--success-color); }
        .bad { color: white; background-color: var(--error-color); }
        .neutral { color: white; background-color: var(--warning-color); }

        .quality-card {
            background: #f8f9fa;
            border-radius: 8px;
```

```
            padding: 15px;
            border-left: 4px solid var(--border-color);
        }

        /* Animation Classes */
        .fade-in { animation: fadeIn 0.5s ease-in; }
        @keyframes fadeIn { from { opacity: 0; transform: translateY(5px);
} to { opacity: 1; transform: translateY(0); } }

        /* Loader */
        .loader {
            display: none;
            text-align: center;
            padding: 20px;
            color: var(--text-secondary);
        }
        .spinner {
            border: 3px solid rgba(0,0,0,0.1);
            border-top: 3px solid var(--primary-color);
            border-radius: 50%;
            width: 24px;
            height: 24px;
            animation: spin 1s linear infinite;
            margin: 0 auto 10px auto;
        }
        @keyframes spin { 0% { transform: rotate(0deg); } 100% { transform:
rotate(360deg); } }

        /* Toast Notifications */
        #toast-container {
            position: fixed;
            bottom: 20px;
            right: 20px;
            z-index: 1000;
            display: flex;
            flex-direction: column;
            gap: 10px;
        }
        .toast {
            background: #333;
            color: #fff;
            padding: 12px 24px;
            border-radius: 8px;
            box-shadow: 0 4px 12px rgba(0,0,0,0.15);
            font-size: 0.9rem;
            animation: slideIn 0.3s ease-out;
            max-width: 300px;
        }
        @keyframes slideIn { from { transform: translateX(100%); opacity:
0; } to { transform: translateX(0); opacity: 1; } }

        /* Empty State */
```

```
252             .empty-state {
253                 text-align: center;
254                 color: #999;
255                 padding: 40px 20px;
256                 font-style: italic;
257             }
258 
259         </style>
260     </head>
261     <body>
262 
263         <div class="main-wrapper">
264             <header>
265                 <h2>Oppia Exploration Editor</h2>
266                 <p>AI-Powered Content Generation & Evaluation Tool</p>
267             </header>
268 
269             <div class="container">
270                 <!-- Left: Content Editor -->
271                 <div class="editor-pane">
272                     <div class="pane-header">
273                         <h3>✏️ State Content</h3>
274                     </div>
275                     <div class="pane-body">
276                         <p style="margin-top:0; font-size: 0.9em; color: #66
6;">Enter the educational text for this state below:</p>
277 
278                         <textarea id="contentInput" placeholder="e.g., Photosyn
thesis is a process used by plants and other organisms to convert light energy into
chemical energy..."></textarea>
279 
280                         <div class="controls">
281                             <button id="btnGen" onclick="triggerGeneration()">
282                                 <span>✨</span> Generate Suggestions
283                             </button>
284                             <button id="btnCheck" onclick="triggerEvaluation
()">
285                                 <span>📊</span> Check Quality
286                             </button>
287                         </div>
288 
289                         <div id="qualityResults"></div>
290                     </div>
291                 </div>
292 
293                 <!-- Right: AI Assistant -->
294                 <div class="ai-pane">
295                     <div class="pane-header">
296                         <h3>🤖 AI Assistant</h3>
297                     </div>
298                     <div class="pane-body" id="aiOutputPane">
299                         <div class="empty-state" id="initialState">
```

```
                            Suggestions and hints will appear here once you gen
erate content.
                        </div>

                        <!-- Loader -->
                        <div class="loader" id="aiLoader">
                            <div class="spinner"></div>
                            <span>Analyzing content...</span>
                        </div>

                        <!-- Output Container -->
                        <div id="suggestions" style="display: none;"></div>
                    </div>
                </div>
            </div>
        </div>

        <!-- Toast Container -->
        <div id="toast-container"></div>

        <script>
            // --- Utility Functions ---

            // Custom Toast Notification (replaces native alert)
            function showToast(message, type = 'info') {
                const container = document.getElementById('toast-container');
                const toast = document.createElement('div');
                toast.className = 'toast';
                toast.textContent = message;

                if(type === 'error') toast.style.borderLeft = '4px solid #F4433
6';
                if(type === 'success') toast.style.borderLeft = '4px solid #4CA
F50';

                container.appendChild(toast);

                // Remove after 3 seconds
                setTimeout(() => {
                    toast.style.opacity = '0';
                    toast.style.transform = 'translateY(10px)';
                    setTimeout(() => container.removeChild(toast), 300);
                }, 3000);
            }

            function setLoading(isLoading) {
                const loader = document.getElementById('aiLoader');
                const suggestions = document.getElementById('suggestions');
                const initialState = document.getElementById('initialState');
                const btnGen = document.getElementById('btnGen');
                const btnCheck = document.getElementById('btnCheck');

```

```
            if (isLoading) {
                loader.style.display = 'block';
                suggestions.style.display = 'none';
                initialState.style.display = 'none';
                btnGen.disabled = true;
                btnCheck.disabled = true;
            } else {
                loader.style.display = 'none';
                suggestions.style.display = 'block';
                btnGen.disabled = false;
                btnCheck.disabled = false;
            }
        }

        // --- Simulated AI Logic (Backend Replacement) ---

        function getKeywords(text) {
            const words = text.split(/\s+/).filter(w => w.length > 4);
            const uniqueWords = [...new Set(words)];
            // Return top 3 longest words as "topics"
            return uniqueWords.sort((a,b) => b.length - a.length).slice(0,
3);
        }

        function simulateAIGeneration(content) {
            return new Promise((resolve) => {
                setTimeout(() => {
                    const topics = getKeywords(content);
                    const mainTopic = topics[0] || "this concept";

                    const data = {
                        questions: [
                            `What are the main characteristics of ${mainTop
ic}?`,
                            `How does ${mainTopic} affect the surrounding e
nvironment?`,
                            `Can you explain the process of ${mainTopic} in
your own words?`
                        ],
                        hint: "Think about the definitions and the key exam
ples provided in the text above.",
                        feedback: "Not quite! Remember to focus on the spec
ific mechanism mentioned in the second paragraph."
                    };
                    resolve(data);
                }, 1200); // Simulate network delay
            });
        }

        function simulateAIEvaluation(content) {
            return new Promise((resolve) => {
                setTimeout(() => {
```

```
                        const wordCount = content.split(/\s+/).length;
                        let score, status, warning, badgeClass;

                        // Simple logic for demonstration
                        if (wordCount < 10) {
                            score = 1;
                            status = "Too Short";
                            badgeClass = "bad";
                            warning = "Content is too brief to be educationa
l.";
                        } else if (wordCount > 100) {
                            score = 12; // High grade level
                            status = "Too Complex";
                            badgeClass = "bad";
                            warning = "Consider breaking this into smaller chun
ks.";
                        } else {
                            score = 5 + Math.floor(Math.random() * 3); // Rando
m grade between 5 and 7
                            status = "Good";
                            badgeClass = "good";
                            warning = null;
                        }

                        const data = {
                            readability_score: score,
                            status: status,
                            warning: warning,
                            badgeClass: badgeClass
                        };
                        resolve(data);
                    }, 1000);
                });
            }

            // --- Main Interaction Functions ---

            async function triggerGeneration() {
                const content = document.getElementById('contentInput').value.t
rim();

                if (!content) {
                    showToast("Please enter some text first.", "error");
                    document.getElementById('contentInput').focus();
                    return;
                }

                setLoading(true);

                try {
                    // In the real app, this would be: const response = await f
etch('/api/generate', ...);
```

```
                    const data = await simulateAIGeneration(content);
                    displaySuggestions(data);
                    showToast("Suggestions generated successfully!", "succes
s");
                } catch (error) {
                    console.error(error);
                    showToast("Failed to generate suggestions.", "error");
                } finally {
                    setLoading(false);
                }
            }

            async function triggerEvaluation() {
                const content = document.getElementById('contentInput').value.t
rim();

                if (!content) {
                    showToast("Please enter some text first.", "error");
                    return;
                }

                // Show a mini loading state specifically for quality if neede
d,
                // but we reuse the main loader for this prototype
                const container = document.getElementById('qualityResults');
                container.innerHTML = `<div style="padding:10px; color:#666;">A
nalyzing quality...</div>`;

                try {
                    const data = await simulateAIEvaluation(content);
                    displayQuality(data);
                    showToast("Quality check complete.", "success");
                } catch (error) {
                    console.error(error);
                    showToast("Failed to evaluate quality.", "error");
                    container.innerHTML = '';
                }
            }

            // --- Display Functions ---

            function displaySuggestions(data) {
                const container = document.getElementById('suggestions');
                let html = '<div class="fade-in">';

                html += '<div class="output-group">';
                html += '<h4>Suggested Questions</h4>';
                if (data.questions && data.questions.length > 0) {
                    data.questions.forEach(q => {
                        const safeQ = q.replace(/'/g, "&#39;");
                        html += `<div class="suggestion-item" onclick="useSugge
stion('${safeQ}')">${q}</div>`;
```

```
                    });
                } else {
                    html += '<p>No questions generated. Try using more descript
ive text.</p>';
                }
                html += '</div>';

                html += '<div class="output-group">';
                html += '<h4>Suggested Hint</h4>';
                html += `<div class="suggestion-item" style="cursor:default; bo
rder-left: 4px solid #FF9800;">${data.hint}</div>`;
                html += '</div>';

                html += '<div class="output-group">';
                html += '<h4>Default Feedback</h4>';
                html += `<div class="suggestion-item" style="cursor:default; bo
rder-left: 4px solid #2196F3;">${data.feedback}</div>`;
                html += '</div>';

                html += '</div>';
                container.innerHTML = html;
            }

            function displayQuality(data) {
                const container = document.getElementById('qualityResults');

                container.innerHTML = `
                    <div class="fade-in">
                        <h4>Quality Check Results</h4>
                        <div class="quality-card" style="border-left-color: ${d
ata.badgeClass === 'good' ? '#4CAF50' : (data.badgeClass === 'bad' ? '#F44336' : '#F
F9800')}">
                            <p style="margin: 0 0 10px 0;"><strong>Readability
Level:</strong> Grade ${data.readability_score}</p>
                            <span class="quality-badge ${data.badgeClass}">${da
ta.status}</span>
                            ${data.warning ? `<p style="color: #d32f2f; font-si
ze: 0.9em; margin-top: 12px; margin-bottom:0;">⚠️ ${data.warning}</p>` : ''}
                        </div>
                    </div>
                `;
            }

            function useSuggestion(text) {
                // In a real integration, this might copy to clipboard or fill
a form field.
                // For this UI demo, we show a nice visual feedback.
                showToast(`Selected: "${text.substring(0, 25)}..."`, "succes
s");

                // Optional: Paste into textarea to show effect
                // document.getElementById('contentInput').value += `\n\nQ: ${t
```

```
ext}`;
            }
        </script>
    </body>
    </html>
```