Business Analyzer Chrome Extension - Complete Development Guide

Project Overview

This Chrome extension will analyze any webpage and generate business ideas, market analysis, competitor research, and actionable startup guides. Perfect for entrepreneurs and business students.

Extension Architecture

Core Components

1. **Manifest** - Extension configuration
2. **Content Script** - Page analysis and data extraction
3. **Popup Interface** - User interaction and results display
4. **Background Service** - API calls and data processing
5. **Export Functionality** - PDF/CSV generation

Step 1: Project Setup

1.1 Create Project Structure

```
business-analyzer-extension/
├── manifest.json
├── popup/
│   ├── popup.html
│   ├── popup.css
│   └── popup.js
├── content/
│   └── content.js
├── background/
│   └── background.js
├── assets/
│   ├── icon16.png
│   ├── icon48.png
│   └── icon128.png
└── utils/
    ├── analyzer.js
    └── export.js
```

1.2 Initialize Manifest (manifest.json)

json

```
{
  "manifest_version": 3,
  "name": "Business Opportunity Analyzer",
  "version": "1.0.0",
  "description": "Analyze web pages for business opportunities and market insights",
  "permissions": [
    "activeTab",
    "storage",
    "scripting"
  ],
  "host_permissions": [
    "https://*/*",
    "http://*/*"
  ],
  "background": {
    "service_worker": "background/background.js"
  },
  "content_scripts": [
    {
      "matches": ["<all_urls>"],
      "js": ["content/content.js"]
    }
  ],
  "action": {
    "default_popup": "popup/popup.html",
    "default_title": "Analyze Business Opportunities",
    "default_icon": {
      "16": "assets/icon16.png",
      "48": "assets/icon48.png",
      "128": "assets/icon128.png"
    }
  },
  "icons": {
    "16": "assets/icon16.png",
    "48": "assets/icon48.png",
    "128": "assets/icon128.png"
  }
}
```

Step 2: Content Script Development

2.1 Page Analysis (content/content.js)

javascript

```
class PageAnalyzer {
  constructor() {
    this.pageData = {};
  }

  extractPageContent() {
    const content = {
      url: window.location.href,
      title: document.title,
      description: this.getMetaDescription(),
      keywords: this.getMetaKeywords(),
      headings: this.extractHeadings(),
      mainContent: this.extractMainContent(),
      images: this.extractImages(),
      links: this.extractLinks(),
      socialMedia: this.extractSocialLinks(),
      contactInfo: this.extractContactInfo(),
      pricing: this.extractPricingInfo(),
      technology: this.detectTechnology()
    };

    return content;
  }

  getMetaDescription() {
    const meta = document.querySelector('meta[name="description"]');
    return meta ? meta.content : '';
  }

  getMetaKeywords() {
    const meta = document.querySelector('meta[name="keywords"]');
    return meta ? meta.content.split(',').map(k => k.trim()) : [];
  }

  extractHeadings() {
    const headings = [];
    const elements = document.querySelectorAll('h1, h2, h3, h4, h5, h6');
    elements.forEach(el => {
      headings.push({
        level: el.tagName.toLowerCase(),
        text: el.textContent.trim()
      });
    });
```

```
    return headings;
  }

  extractMainContent() {
    // Remove navigation, footer, sidebar elements
    const elementsToRemove = 'nav, footer, aside, .sidebar, .menu, .navigation';
    const clonedDoc = document.cloneNode(true);
    clonedDoc.querySelectorAll(elementsToRemove).forEach(el => el.remove());

    const mainContent = clonedDoc.querySelector('main, article, .content, .main-content')
      || clonedDoc.body;

    return mainContent.textContent.trim().substring(0, 2000);
  }

  extractImages() {
    const images = [];
    document.querySelectorAll('img').forEach(img => {
      if (img.src && img.alt) {
        images.push({
          src: img.src,
          alt: img.alt
        });
      }
    });
    return images.slice(0, 10); // Limit to 10 images
  }

  extractLinks() {
    const links = [];
    document.querySelectorAll('a[href]').forEach(link => {
      const href = link.href;
      if (href.startsWith('http')) {
        links.push({
          url: href,
          text: link.textContent.trim(),
          domain: new URL(href).hostname
        });
      }
    });
    return links.slice(0, 20); // Limit to 20 links
  }

  extractSocialLinks() {
```

```
    const socialPlatforms = ['facebook', 'twitter', 'linkedin', 'instagram', 'youtube', 'tiktok'];
    const socialLinks = [];

    document.querySelectorAll('a[href]').forEach(link => {
      const href = link.href.toLowerCase();
      socialPlatforms.forEach(platform => {
        if (href.includes(platform)) {
          socialLinks.push({
            platform: platform,
            url: link.href
          });
        }
      });
    });

    return socialLinks;
  }

  extractContactInfo() {
    const emailRegex = /\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b/g;
    const phoneRegex = /(\+?1?[-.\s]?)?\(?([0-9]{3})\)?[-.\s]?([0-9]{3})[-.\s]?([0-9]{4})/g;

    const bodyText = document.body.textContent;
    const emails = bodyText.match(emailRegex) || [];
    const phones = bodyText.match(phoneRegex) || [];

    return {
      emails: [...new Set(emails)].slice(0, 5),
      phones: [...new Set(phones)].slice(0, 5)
    };
  }

  extractPricingInfo() {
    const priceRegex = /\$[\d,]+\.?\d*/g;
    const bodyText = document.body.textContent;
    const prices = bodyText.match(priceRegex) || [];

    return [...new Set(prices)].slice(0, 10);
  }

  detectTechnology() {
    const technologies = [];

    // Check for common frameworks and libraries
```

```
    if (window.React) technologies.push('React');
    if (window.Vue) technologies.push('Vue.js');
    if (window.angular) technologies.push('Angular');
    if (window.jQuery || window.$) technologies.push('jQuery');

    // Check meta tags and script sources
    document.querySelectorAll('script[src]').forEach(script => {
      const src = script.src.toLowerCase();
      if (src.includes('shopify')) technologies.push('Shopify');
      if (src.includes('wordpress')) technologies.push('WordPress');
      if (src.includes('wix')) technologies.push('Wix');
    });

    return technologies;
  }
}

// Listen for messages from popup
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {
  if (request.action === 'analyzeCurrentPage') {
    const analyzer = new PageAnalyzer();
    const pageData = analyzer.extractPageContent();
    sendResponse({ success: true, data: pageData });
  }
});
```

Step 3: Background Service Worker

3.1 API Integration (background/background.js)

javascript

```
class BusinessAnalyzer {
  constructor() {
    this.apiEndpoints = {
      openai: 'https://api.openai.com/v1/chat/completions',
      // You can add other AI services here
    };
  }

  async generateBusinessIdeas(pageData) {
    const prompt = this.createBusinessIdeaPrompt(pageData);

    try {
      const response = await this.callOpenAI(prompt, 'business-ideas');
      return this.parseBusinessIdeasResponse(response);
    } catch (error) {
      console.error('Error generating business ideas:', error);
      return this.generateFallbackIdeas(pageData);
    }
  }

  createBusinessIdeaPrompt(pageData) {
    return `
Analyze this webpage data and generate 5 specific business ideas:

Website: ${pageData.title}
URL: ${pageData.url}
Description: ${pageData.description}
Main Content Preview: ${pageData.mainContent}
Industry Indicators: ${pageData.keywords?.join(', ')}
Technology Stack: ${pageData.technology?.join(', ')}
Pricing Information: ${pageData.pricing?.join(', ')}

For each business idea, provide:
1. Idea name and brief description
2. Target market and customer segment
3. Revenue model
4. Market size estimate
5. Key competitors (3-5)
6. Startup difficulty (1-10 scale)
7. Initial investment required
8. Time to first customer estimate

Format as structured JSON with these exact keys: ideas[{name, description, targetMarket, revenueModel, marketSize, co
```

```
`;
  }

  async callOpenAI(prompt, context) {
    // Note: In production, API key should be stored securely
    const apiKey = await this.getStoredApiKey();

    const response = await fetch(this.apiEndpoints.openai, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${apiKey}`
      },
      body: JSON.stringify({
        model: 'gpt-4',
        messages: [
          {
            role: 'system',
            content: 'You are a business analyst expert who identifies opportunities and provides actionable insights.'
          },
          {
            role: 'user',
            content: prompt
          }
        ],
        max_tokens: 2000,
        temperature: 0.7
      })
    });

    const data = await response.json();
    return data.choices[0].message.content;
  }

  parseBusinessIdeasResponse(response) {
    try {
      // Extract JSON from response
      const jsonMatch = response.match(/\{[\s\S]*\}/);
      if (jsonMatch) {
        return JSON.parse(jsonMatch[0]);
      }
      throw new Error('No JSON found in response');
    } catch (error) {
      console.error('Error parsing AI response:', error);
```

```
      return this.generateFallbackIdeas();
    }
  }

  generateFallbackIdeas(pageData = {}) {
    return {
      ideas: [
        {
          name: "Competitive Analysis Service",
          description: "Provide detailed competitor research for businesses in similar industries",
          targetMarket: "Small to medium businesses",
          revenueModel: "Monthly subscription + custom reports",
          marketSize: "$2B+ market research industry",
          competitors: ["SEMrush", "SimilarWeb", "Ahrefs"],
          difficulty: 6,
          investment: "$10,000 - $25,000",
          timeToCustomer: "30-45 days"
        },
        {
          name: "Industry Newsletter & Insights",
          description: "Curated newsletter with industry trends and opportunities",
          targetMarket: "Industry professionals and entrepreneurs",
          revenueModel: "Subscription + sponsored content",
          marketSize: "$1.5B newsletter market",
          competitors: ["Morning Brew", "The Hustle", "Substack"],
          difficulty: 4,
          investment: "$1,000 - $5,000",
          timeToCustomer: "14-21 days"
        }
      ]
    };
  }

  async generateStartupGuide(businessIdea) {
    const prompt = `
Create a detailed step-by-step startup guide for this business idea:

Business: ${businessIdea.name}
Description: ${businessIdea.description}
Target Market: ${businessIdea.targetMarket}
Revenue Model: ${businessIdea.revenueModel}

Provide a comprehensive 10-step guide covering:
1. Market validation
```

```
2. Business planning
3. Legal setup
4. Initial funding
5. Product/service development
6. Brand and marketing
7. Sales strategy
8. Operations setup
9. Finding first customers
10. Scaling strategies

Format as JSON with steps[{stepNumber, title, description, timeframe, cost, resources[]}]
`;

    try {
      const response = await this.callOpenAI(prompt, 'startup-guide');
      return this.parseStartupGuideResponse(response);
    } catch (error) {
      return this.generateFallbackStartupGuide(businessIdea);
    }
  }

  parseStartupGuideResponse(response) {
    try {
      const jsonMatch = response.match(/\{[\s\S]*\}/);
      if (jsonMatch) {
        return JSON.parse(jsonMatch[0]);
      }
      throw new Error('No JSON found in response');
    } catch (error) {
      return this.generateFallbackStartupGuide();
    }
  }

  generateFallbackStartupGuide(idea) {
    return {
      steps: [
        {
          stepNumber: 1,
          title: "Validate Your Business Idea",
          description: "Conduct market research and validate demand through surveys, interviews, and landing page tests.",
          timeframe: "2-4 weeks",
          cost: "$500-$1,500",
          resources: ["Google Forms", "Typeform", "UserInterviews.com", "Landing page builders"]
        },
```

```
        {
          stepNumber: 2,
          title: "Create Business Plan",
          description: "Develop a comprehensive business plan including financial projections and go-to-market strategy.",
          timeframe: "1-2 weeks",
          cost: "$0-$500",
          resources: ["Business plan templates", "Financial modeling tools", "SCORE mentoring"]
        },
        // Add remaining 8 steps...
      ]
    };
  }

  async getStoredApiKey() {
    return new Promise((resolve) => {
      chrome.storage.sync.get(['openai_api_key'], (result) => {
        resolve(result.openai_api_key || '');
      });
    });
  }
}

// Message handling
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {
  const analyzer = new BusinessAnalyzer();

  if (request.action === 'generateBusinessIdeas') {
    analyzer.generateBusinessIdeas(request.pageData)
      .then(ideas => sendResponse({ success: true, data: ideas }))
      .catch(error => sendResponse({ success: false, error: error.message }));
    return true; // Async response
  }

  if (request.action === 'generateStartupGuide') {
    analyzer.generateStartupGuide(request.businessIdea)
      .then(guide => sendResponse({ success: true, data: guide }))
      .catch(error => sendResponse({ success: false, error: error.message }));
    return true; // Async response
  }
});
```

Step 4: Popup Interface

4.1 HTML Structure (popup/popup.html)

```html
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Business Analyzer</title>
    <link rel="stylesheet" href="popup.css">
</head>
<body>
    <div class="container">
        <header>
            <h1>🚀 Business Opportunity Analyzer</h1>
            <div id="status" class="status">Ready to analyze</div>
        </header>

        <div id="settings-section" class="section">
            <h3>Settings</h3>
            <input type="password" id="api-key" placeholder="Enter OpenAI API Key">
            <button id="save-key">Save Key</button>
        </div>

        <div class="section">
            <button id="analyze-btn" class="primary-btn">
                Analyze Current Page
            </button>
        </div>

        <div id="results-section" class="section hidden">
            <h3>Business Opportunities</h3>
            <div id="business-ideas"></div>
        </div>

        <div id="startup-guide-section" class="section hidden">
            <h3>Startup Guide</h3>
            <div id="startup-guide"></div>
        </div>

        <div id="export-section" class="section hidden">
            <button id="export-pdf" class="export-btn">Export as PDF</button>
            <button id="export-csv" class="export-btn">Export as CSV</button>
        </div>

        <div id="loading" class="loading hidden">
```

```
        <div class="spinner"></div>
        <p>Analyzing page and generating insights...</p>
      </div>
    </div>

    <script src="../utils/export.js"></script>
    <script src="popup.js"></script>
</body>
</html>
```

4.2 CSS Styling (popup/popup.css)

```
css
```

```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    width: 400px;
    min-height: 500px;
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: #333;
}

.container {
    padding: 20px;
}

header {
    text-align: center;
    margin-bottom: 20px;
}

header h1 {
    color: white;
    font-size: 18px;
    margin-bottom: 8px;
}

.status {
    background: rgba(255, 255, 255, 0.2);
    color: white;
    padding: 5px 10px;
    border-radius: 15px;
    font-size: 12px;
    display: inline-block;
}

.section {
    background: white;
    border-radius: 8px;
    padding: 15px;
    margin-bottom: 15px;
```

```
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

.section h3 {
    color: #4a5568;
    margin-bottom: 10px;
    font-size: 14px;
    font-weight: 600;
}

.primary-btn {
    width: 100%;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    border: none;
    padding: 12px;
    border-radius: 6px;
    cursor: pointer;
    font-weight: 600;
    transition: transform 0.2s;
}

.primary-btn:hover {
    transform: translateY(-2px);
}

.export-btn {
    background: #48bb78;
    color: white;
    border: none;
    padding: 8px 16px;
    border-radius: 4px;
    cursor: pointer;
    margin-right: 10px;
    margin-bottom: 5px;
}

.business-idea {
    border: 1px solid #e2e8f0;
    border-radius: 6px;
    padding: 15px;
    margin-bottom: 15px;
    transition: box-shadow 0.2s;
}
```

```css
.business-idea:hover {
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
}

.business-idea h4 {
    color: #2d3748;
    margin-bottom: 8px;
    cursor: pointer;
}

.business-idea h4:hover {
    color: #667eea;
}

.idea-meta {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 10px;
    margin-top: 10px;
    font-size: 12px;
}

.meta-item {
    background: #f7fafc;
    padding: 5px 8px;
    border-radius: 4px;
}

.meta-label {
    font-weight: 600;
    color: #4a5568;
}

.competitors {
    margin-top: 8px;
}

.competitor-tag {
    display: inline-block;
    background: #e2e8f0;
    padding: 2px 8px;
    border-radius: 12px;
    font-size: 10px;
```

```
    margin-right: 5px;
    margin-bottom: 3px;
}

.startup-step {
    border-left: 3px solid #667eea;
    padding: 10px 15px;
    margin-bottom: 15px;
    background: #f8fafc;
}

.step-header {
    display: flex;
    align-items: center;
    margin-bottom: 8px;
}

.step-number {
    background: #667eea;
    color: white;
    width: 24px;
    height: 24px;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 12px;
    font-weight: bold;
    margin-right: 10px;
}

.step-title {
    font-weight: 600;
    color: #2d3748;
}

.step-meta {
    display: flex;
    gap: 15px;
    margin-top: 8px;
    font-size: 11px;
    color: #666;
}
```

```css
.hidden {
    display: none !important;
}

.loading {
    text-align: center;
    padding: 30px;
}

.spinner {
    width: 40px;
    height: 40px;
    border: 4px solid #f3f3f3;
    border-top: 4px solid #667eea;
    border-radius: 50%;
    animation: spin 1s linear infinite;
    margin: 0 auto 15px;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}

#api-key {
    width: 70%;
    padding: 8px;
    border: 1px solid #ddd;
    border-radius: 4px;
    margin-right: 8px;
}

#save-key {
    background: #48bb78;
    color: white;
    border: none;
    padding: 8px 12px;
    border-radius: 4px;
    cursor: pointer;
}
```

4.3 Popup JavaScript (popup/popup.js)

javascript

```
class PopupController {
  constructor() {
    this.currentBusinessIdeas = null;
    this.currentStartupGuide = null;
    this.initializeEventListeners();
    this.loadSavedApiKey();
  }

  initializeEventListeners() {
    document.getElementById('save-key').addEventListener('click', () => this.saveApiKey());
    document.getElementById('analyze-btn').addEventListener('click', () => this.analyzeCurrentPage());
    document.getElementById('export-pdf').addEventListener('click', () => this.exportToPDF());
    document.getElementById('export-csv').addEventListener('click', () => this.exportToCSV());
  }

  async loadSavedApiKey() {
    const result = await chrome.storage.sync.get(['openai_api_key']);
    if (result.openai_api_key) {
      document.getElementById('api-key').value = result.openai_api_key;
    }
  }

  async saveApiKey() {
    const apiKey = document.getElementById('api-key').value;
    if (apiKey) {
      await chrome.storage.sync.set({ openai_api_key: apiKey });
      this.updateStatus('API key saved successfully', 'success');
    }
  }

  async analyzeCurrentPage() {
    this.showLoading(true);
    this.updateStatus('Analyzing current page...', 'loading');

    try {
      // Get current tab
      const [tab] = await chrome.tabs.query({ active: true, currentWindow: true });

      // Extract page data using content script
      const pageDataResponse = await chrome.tabs.sendMessage(tab.id, {
        action: 'analyzeCurrentPage'
      });
```

```
      if (!pageDataResponse.success) {
        throw new Error('Failed to extract page data');
      }

      // Generate business ideas
      const businessIdeasResponse = await chrome.runtime.sendMessage({
        action: 'generateBusinessIdeas',
        pageData: pageDataResponse.data
      });

      if (!businessIdeasResponse.success) {
        throw new Error(businessIdeasResponse.error || 'Failed to generate business ideas');
      }

      this.currentBusinessIdeas = businessIdeasResponse.data;
      this.displayBusinessIdeas(this.currentBusinessIdeas);
      this.updateStatus('Analysis complete', 'success');

    } catch (error) {
      console.error('Analysis error:', error);
      this.updateStatus('Analysis failed: ' + error.message, 'error');
    } finally {
      this.showLoading(false);
    }
  }

  displayBusinessIdeas(ideas) {
    const container = document.getElementById('business-ideas');
    container.innerHTML = '';

    ideas.ideas.forEach((idea, index) => {
      const ideaElement = this.createBusinessIdeaElement(idea, index);
      container.appendChild(ideaElement);
    });

    document.getElementById('results-section').classList.remove('hidden');
    document.getElementById('export-section').classList.remove('hidden');
  }

  createBusinessIdeaElement(idea, index) {
    const div = document.createElement('div');
    div.className = 'business-idea';
    div.innerHTML = `
      <h4 onclick="popupController.showStartupGuide(${index})">${idea.name}</h4>
```

```
        <p>${idea.description}</p>

        <div class="idea-meta">
          <div class="meta-item">
            <div class="meta-label">Target Market</div>
            <div>${idea.targetMarket}</div>
          </div>
          <div class="meta-item">
            <div class="meta-label">Revenue Model</div>
            <div>${idea.revenueModel}</div>
          </div>
          <div class="meta-item">
            <div class="meta-label">Market Size</div>
            <div>${idea.marketSize}</div>
          </div>
          <div class="meta-item">
            <div class="meta-label">Difficulty</div>
            <div>${idea.difficulty}/10</div>
          </div>
          <div class="meta-item">
            <div class="meta-label">Investment</div>
            <div>${idea.investment}</div>
          </div>
          <div class="meta-item">
            <div class="meta-label">Time to Customer</div>
            <div>${idea.timeToCustomer}</div>
          </div>
        </div>

        <div class="competitors">
          <div class="meta-label">Key Competitors:</div>
          ${idea.competitors.map(comp => `<span class="competitor-tag">${comp}</span>`).join('')}
        </div>
      `;
      return div;
    }

    async showStartupGuide(ideaIndex) {
      if (!this.currentBusinessIdeas || !this.currentBusinessIdeas.ideas[ideaIndex]) {
        return;
      }

      const businessIdea = this.currentBusinessIdeas.ideas[ideaIndex];
      this.showLoading(true);
```

```
    this.updateStatus('Generating startup guide...', 'loading');

    try {
      const response = await chrome.runtime.sendMessage({
        action: 'generateStartupGuide',
        businessIdea: businessIdea
      });

      if (!response.success) {
        throw new Error(response.error || 'Failed to generate startup guide');
      }

      this.currentStartupGuide = response.data;
      this.displayStartupGuide(this.currentStartupGuide);
      this.updateStatus('Startup guide generated', 'success');

    } catch (error) {
      console.error('Startup guide error:', error);
      this.updateStatus('Failed to generate guide: ' + error.message, 'error');
    } finally {
      this.showLoading(false);
    }
  }

  displayStartupGuide(guide) {
    const container = document.getElementById('startup-guide');
    container.innerHTML = '';

    guide.steps.forEach(step => {
      const stepElement = this.createStartupStepElement(step);
      container.appendChild(stepElement);
    });

    document.getElementById('startup-guide-section').classList.remove('hidden');

    // Scroll to startup guide
    document.getElementById('startup-guide-section').scrollIntoView({
      behavior: 'smooth',
      block: 'start'
    });
  }

  createStartupStepElement(step) {
    const div = document.createElement('div');
```

```
        div.className = 'startup-step';
        div.innerHTML = `
            <div class="step-header">
                <div class="step-number">${step.stepNumber}</div>
                <div class="step-title">${step.title}</div>
            </div>
            <div class="step-description">${step.description}</div>
            <div class="step-meta">
                <span><strong>Timeframe:</strong> ${step.timeframe}</span>
                <span><strong>Cost:</strong> ${step.cost}</span>
            </div>
            <div class="step-resources">
                <div class="meta-label">Resources:</div>
                ${step.resources.map(resource => `<span class="competitor-tag">${resource}</span>`).join('')}
            </div>
        `;
        return div;
    }

    exportToPDF() {
        if (!this.currentBusinessIdeas) return;

        const exporter = new ExportUtils();
        exporter.exportToPDF({
            businessIdeas: this.currentBusinessIdeas,
            startupGuide: this.currentStartupGuide
        });
    }

    exportToCSV() {
        if (!this.currentBusinessIdeas) return;

        const exporter = new ExportUtils();
        exporter.exportToCSV(this.currentBusinessIdeas);
    }

    showLoading(show) {
        const loadingElement = document.getElementById('loading');
        if (show) {
            loadingElement.classList.remove('hidden');
        } else {
            loadingElement.classList.add('hidden');
        }
    }
```

```
  updateStatus(message, type = 'info') {
    const statusElement = document.getElementById('status');
    statusElement.textContent = message;

    // Remove existing status classes
    statusElement.classList.remove('status-success', 'status-error', 'status-loading');

    // Add appropriate class
    if (type === 'success') {
      statusElement.classList.add('status-success');
    } else if (type === 'error') {
      statusElement.classList.add('status-error');
    } else if (type === 'loading') {
      statusElement.classList.add('status-loading');
    }
  }
}

// Initialize popup controller
const popupController = new PopupController();
```

Step 5: Export Functionality

5.1 Export Utilities (utils/export.js)

javascript

```
class ExportUtils {
  constructor() {
    this.currentDate = new Date().toISOString().split('T')[0];
  }

  exportToPDF(data) {
    // Create HTML content for PDF
    const htmlContent = this.generatePDFContent(data);

    // Create a new window for PDF generation
    const printWindow = window.open('', '_blank');
    printWindow.document.write(htmlContent);
    printWindow.document.close();

    // Trigger print dialog
    printWindow.onload = () => {
      printWindow.print();
      setTimeout(() => printWindow.close(), 1000);
    };
  }

  generatePDFContent(data) {
    const { businessIdeas, startupGuide } = data;

    let html = `
<!DOCTYPE html>
<html>
<head>
    <title>Business Analysis Report</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; line-height: 1.6; }
        .header { text-align: center; margin-bottom: 30px; }
        .business-idea { margin-bottom: 30px; border: 1px solid #ddd; padding: 15px; }
        .idea-title { color: #333; font-size: 18px; font-weight: bold; margin-bottom: 10px; }
        .meta-grid { display: grid; grid-template-columns: 1fr 1fr; gap: 10px; margin: 15px 0; }
        .meta-item { background: #f5f5f5; padding: 8px; }
        .competitors { margin-top: 10px; }
        .competitor-tag { background: #e0e0e0; padding: 2px 8px; margin: 2px; border-radius: 10px; }
        .startup-step { margin-bottom: 20px; border-left: 3px solid #007cba; padding-left: 15px; }
        .step-number { background: #007cba; color: white; width: 25px; height: 25px; border-radius: 50%; display: inline-fle
        @media print {
            .business-idea { page-break-inside: avoid; }
            .startup-step { page-break-inside: avoid; }
```

```
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>🚀 Business Analysis Report</h1>
        <p>Generated on ${new Date().toLocaleDateString()}</p>
        <p>URL: ${window.location.href}</p>
    </div>

    <h2>Business Opportunities</h2>
`;

    businessIdeas.ideas.forEach((idea, index) => {
        html += `
        <div class="business-idea">
            <div class="idea-title">${idea.name}</div>
            <p><strong>Description:</strong> ${idea.description}</p>

            <div class="meta-grid">
                <div class="meta-item"><strong>Target Market:</strong> ${idea.targetMarket}</div>
                <div class="meta-item"><strong>Revenue Model:</strong> ${idea.revenueModel}</div>
                <div class="meta-item"><strong>Market Size:</strong> ${idea.marketSize}</div>
                <div class="meta-item"><strong>Difficulty:</strong> ${idea.difficulty}/10</div>
                <div class="meta-item"><strong>Investment:</strong> ${idea.investment}</div>
                <div class="meta-item"><strong>Time to Customer:</strong> ${idea.timeToCustomer}</div>
            </div>

            <div class="competitors">
                <strong>Key Competitors:</strong>
                ${idea.competitors.map(comp => `<span class="competitor-tag">${comp}</span>`).join('')}
            </div>
        </div>
        `;
    });

    if (startupGuide && startupGuide.steps) {
        html += `
        <h2>Startup Guide</h2>
        `;

        startupGuide.steps.forEach(step => {
            html += `
            <div class="startup-step">
```

```
            <div style="display: flex; align-items: center; margin-bottom: 10px;">
                <span class="step-number">${step.stepNumber}</span>
                <strong>${step.title}</strong>
            </div>
            <p>${step.description}</p>
            <p><strong>Timeframe:</strong> ${step.timeframe} | <strong>Cost:</strong> ${step.cost}</p>
            <p><strong>Resources:</strong> ${step.resources.join(', ')}</p>
        </div>
        `;
    });
  }

  html += `
</body>
</html>
  `;

  return html;
 }

 exportToCSV(businessIdeas) {
  const csvData = this.convertToCSV(businessIdeas);
  this.downloadCSV(csvData, `business-analysis-${this.currentDate}.csv`);
 }

 convertToCSV(businessIdeas) {
  const headers = [
   'Business Name',
   'Description',
   'Target Market',
   'Revenue Model',
   'Market Size',
   'Competitors',
   'Difficulty (1-10)',
   'Investment Required',
   'Time to First Customer'
  ];

  const rows = businessIdeas.ideas.map(idea => [
   idea.name,
   idea.description,
   idea.targetMarket,
   idea.revenueModel,
   idea.marketSize,
```

```
      idea.competitors.join('; '),
      idea.difficulty,
      idea.investment,
      idea.timeToCustomer
    ]);

    const csvContent = [headers, ...rows]
      .map(row => row.map(cell => `"${cell.toString().replace(/"/g, '""')}"`).join(','))
      .join('\n');

    return csvContent;
  }

  downloadCSV(csvContent, filename) {
    const blob = new Blob([csvContent], { type: 'text/csv' });
    const url = URL.createObjectURL(blob);

    const a = document.createElement('a');
    a.href = url;
    a.download = filename;
    document.body.appendChild(a);
    a.click();
    document.body.removeChild(a);
    URL.revokeObjectURL(url);
  }
}
```

Step 6: Additional CSS Improvements

6.1 Enhanced Status Styles (popup/popup.css - add to existing)

```
css
```

```
.status-success {
    background: rgba(72, 187, 120, 0.2) !important;
    color: #22543d !important;
}

.status-error {
    background: rgba(245, 101, 101, 0.2) !important;
    color: #742a2a !important;
}

.status-loading {
    background: rgba(66, 153, 225, 0.2) !important;
    color: #2a4365 !important;
}

.fade-in {
    animation: fadeIn 0.5s ease-in;
}

@keyframes fadeIn {
    from { opacity: 0; transform: translateY(10px); }
    to { opacity: 1; transform: translateY(0); }
}

.export-section {
    text-align: center;
}

.export-btn:hover {
    background: #38a169;
    transform: translateY(-1px);
}
```

Step 7: Installation and Testing

7.1 Extension Installation Steps

1. **Package the Extension**

```
bash
```

```
# Create the project folder structure
mkdir business-analyzer-extension
cd business-analyzer-extension
# Add all the files as described above
```

2. **Load in Chrome**
   - Open Chrome and go to `chrome://extensions/`
   - Enable "Developer mode" (top right toggle)
   - Click "Load unpacked"
   - Select your extension folder
3. **Test Basic Functionality**
   - Navigate to any website
   - Click the extension icon
   - Enter your OpenAI API key in settings
   - Click "Analyze Current Page"

7.2 Testing Checklist

- ✅ Extension loads without errors
- ✅ Content script extracts page data
- ✅ API key is saved and retrieved
- ✅ Business ideas are generated and displayed
- ✅ Startup guides are generated when clicking business ideas
- ✅ Export to PDF works
- ✅ Export to CSV works
- ✅ Error handling works for failed API calls

Step 8: Advanced Features & Enhancements

8.1 Future Improvements

1. **Multiple AI Providers**: Add support for Anthropic Claude, Google Gemini
2. **Local Storage**: Save analysis history
3. **Competitor Research**: Integrate with tools like SimilarWeb
4. **Market Data**: Connect to industry databases
5. **Social Media Analysis**: Extract social engagement data

6. **SEO Insights**: Add keyword and traffic analysis
7. **Financial Modeling**: Build revenue projections
8. **Team Collaboration**: Share findings with team members

8.2 Monetization Opportunities

1. **Premium Features**: Advanced AI models, deeper analysis
2. **API Integration**: Premium data sources
3. **Business Templates**: Industry-specific business plan templates
4. **Consulting Services**: Expert review of generated ideas
5. **Community Features**: Rate and share business ideas

Step 9: Deployment and Distribution

9.1 Chrome Web Store Preparation

1. **Create Icons**: Design 16x16, 48x48, 128x128 px icons
2. **Write Store Listing**:
   - Compelling title and description
   - Screenshots of the extension in action
   - Privacy policy and terms of service
3. **Package for Store**: Create .zip file with all extension files
4. **Submit for Review**: Follow Chrome Web Store guidelines

9.2 Privacy and Security

- **API Key Security**: Store encrypted, allow users to use their own keys
- **Data Privacy**: Don't store sensitive page data
- **Permissions**: Request minimal necessary permissions
- **Content Security**: Validate all inputs and outputs

Conclusion

This Chrome extension provides a comprehensive business analysis tool that:

- Analyzes any webpage for business opportunities
- Generates AI-powered business ideas with market insights
- Provides detailed startup guides with actionable steps

- Offers professional export capabilities
- Maintains user privacy and security

The modular architecture allows for easy expansion and integration of additional features. The extension serves as both a learning tool for understanding web extension development and a practical business research assistant.