MANIFEST.JSON:

```
{
  "manifest_version": 3,
  "name": "Social Media Content Filter",
  "version": "1.0",
  "description": "Filtreaza continutul nepotrivit din social media",
  "permissions": [
   "activeTab",
   "storage"
  ],
  "action": {
   "default_popup": "popup.html",
   "default_icon": {
    "16": "icons/icon16.png",
    "48": "icons/icon48.png",
    "128": "icons/icon128.png"
   }
  },
  "content_scripts": [
   {
    "matches": ["*://*.facebook.com/*", "*://*.twitter.com/*", "*://*.instagram.com/*"],
    "js": ["contentFilter.js"]
   }
  ]
 }
```

POPUP.HTML:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      width: 300px;
      padding: 10px;
    }
    .filter-option {
      margin: 10px 0;
    }
  </style>
</head>
<body>
  <h2>Setări Filtru</h2>
  <div class="filter-option">
    <label>
      <input type="checkbox" id="inappropriate" checked>
      Filtreaza continutul nepotrivit
    </label>
  </div>
  <div class="filter-option">
    <label>
      <input type="checkbox" id="spam" checked>
      Filtreaza spam
    </label>
  </div>
  <div class="filter-option">
    <label>
      <input type="checkbox" id="promotional" checked>
      Filtreaza continutul promotional
    </label>
  </div>
  <div class="filter-option">
    <label>
```

```
      <input type="checkbox" id="sensitive" checked>
      Filtreaza continutul sensibil
    </label>
  </div>
  <script src="popup.js"></script>
</body>
</html>
```

POPUP.JS:

```
document.addEventListener('DOMContentLoaded', () => {
    // Încărcarea setărilor salvate
    chrome.storage.sync.get('filterSettings', (data) => {
        const settings = data.filterSettings || {
            inappropriate: true,
            spam: true,
            promotional: true,
            sensitive: true
        };

        // Actualizarea checkbox-urilor
        Object.keys(settings).forEach(key => {
            document.getElementById(key).checked = settings[key];
        });
    });

    // Adăugarea event listeners pentru checkbox-uri
    const checkboxes = document.querySelectorAll('input[type="checkbox"]');
    checkboxes.forEach(checkbox => {
        checkbox.addEventListener('change', () => {
            const settings = {};
            checkboxes.forEach(cb => {
                settings[cb.id] = cb.checked;
            });

            // Salvarea setărilor
            chrome.storage.sync.set({
                filterSettings: settings
            });

            // Notificarea content script-ului
            chrome.tabs.query({active: true, currentWindow: true}, (tabs) => {
                chrome.tabs.sendMessage(tabs[0].id, {
                    type: 'settingsUpdated',
                    settings: settings
                });
            });
        });
    });
});
```

CONTENTFILTER.JS:

```
class SocialMediaProtection {
    constructor() {
        this.filters = {
            inappropriate: true,
            spam: true,
            promotional: true,
            sensitive: true
        };

        this.loadSettings();
        this.initializeObserver();
    }

    // Încărcarea setărilor salvate
    async loadSettings() {
        const settings = await chrome.storage.sync.get('filterSettings');
        if (settings.filterSettings) {
            this.filters = {...this.filters, ...settings.filterSettings};
        }
    }
```

```
// Inițializarea observatorului pentru modificări DOM
initializeObserver() {
  const observer = new MutationObserver((mutations) => {
    mutations.forEach((mutation) => {
      mutation.addedNodes.forEach((node) => {
        if (node.nodeType === 1) { // Element node
          this.processNode(node);
        }
      });
    });
  });

  observer.observe(document.body, {
    childList: true,
    subtree: true
  });
}

// Procesarea conținutului
processNode(node) {
  const textContent = node.textContent;

  if (this.shouldFilter(textContent)) {
    this.applyFilter(node);
  }
}

// Verificarea dacă conținutul trebuie filtrat
shouldFilter(text) {
  if (this.filters.inappropriate && this.containsInappropriateContent(text)) {
    return true;
  }
  if (this.filters.spam && this.isSpam(text)) {
    return true;
  }
  if (this.filters.promotional && this.isPromotional(text)) {
    return true;
    }
    if (this.filters.sensitive && this.containsSensitiveContent(text)) {
        return true;
    }
    return false;
}

// Filtre specifice
containsInappropriateContent(text) {
    const inappropriatePattern = /\b(cuvant1|cuvant2|injurie)\b/i;
    return inappropriatePattern.test(text);
}

isSpam(text) {
    const spamPatterns = [
        /câștigă bani rapid/i,
        /\d{1,3}[%]\s*(?:discount|reducere)/,
        /click aici|accesează acum/i
    ];
    return spamPatterns.some(pattern => pattern.test(text));
}

isPromotional(text) {
    const promoPatterns = [
        /promotie|oferta/i,
        /doar azi|doar acum/i,
        /pret special|pret redus/i
    ];
    return promoPatterns.some(pattern => pattern.test(text));
}

containsSensitiveContent(text) {
    const sensitivePatterns = [
        /violență|agresiune/i,
        /conținut explicit/i,
        /\b(adult|nsfw)\b/i
    ];
```

```
        return sensitivePatterns.some(pattern => pattern.test(text));
    }

    // Aplicarea filtrului
    applyFilter(node) {
        const originalContent = node.innerHTML;

        // Crearea unui overlay de avertizare
        const warning = document.createElement('div');
        warning.className = 'content-warning';
        warning.innerHTML = `
            <div class="warning-message">
                Conținut filtrat
                <button class="show-content">Arată conținutul</button>
            </div>
        `;

        // Salvarea conținutului original
        node.setAttribute('data-original-content', originalContent);
        node.innerHTML = '';
        node.appendChild(warning);

        // Adăugarea eventListener pentru butonul de afișare
        warning.querySelector('.show-content').addEventListener('click', () => {
            node.innerHTML = node.getAttribute('data-original-content');
        });
    }
}

// Stiluri CSS pentru avertizări
const styles = `
    .content-warning {
        background: #f8d7da;
        border: 1px solid #f5c6cb;
        padding: 10px;
        margin: 5px 0;
        border-radius: 4px;
    }
    .warning-message {
        color: #721c24;
        font-size: 14px;
    }
    .show-content {
        background: #dc3545;
        color: white;
        border: none;
        padding: 5px 10px;
        border-radius: 3px;
        margin-left: 10px;
        cursor: pointer;
    }
`;

// Injectarea stilurilor
const styleSheet = document.createElement('style');
styleSheet.textContent = styles;
document.head.appendChild(styleSheet);

// Inițializarea protecției
new SocialMediaProtection();
```