

SECTION B: PRACTICAL

1. Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Opportunity Board 8480</title>

  <link rel="stylesheet" href="styles.css">

</head>

<body>


  <header class="site-header">

    <h1>Opportunity Board 8480</h1>

    <span id="opportunity-counter">0 opportunities found</span>

  </header>


  <nav class="filter-nav">

    <div class="filter-controls">

      <button class="filter-btn active" data-filter="all">All</button>

      <button class="filter-btn" data-filter="internship">Internships</button>

      <button class="filter-btn" data-filter="scholarship">Scholarships</button>

    </div>

  </nav>


  <main class="content-area">

    <section class="opportunity-list">

      <h2>Available Opportunities</h2>

      <div class="add-opportunity-form">
```

```
<label for="opportunity-title-input" class="sr-only">New opportunity title</label>
<input type="text" id="opportunity-title-input" placeholder="Enter opportunity title...">
```

```
<label for="opportunity-type-input" class="sr-only">Opportunity type</label>
<select id="opportunity-type-input">
  <option value="internship">Internship</option>
  <option value="scholarship">Scholarship</option>
</select>
```

```
<button id="add-opportunity-btn">Add</button>
```

```
</div>
```

```
<div id="error-message" class="error-message" role="alert"></div>
```

```
<div id="available-list" class="card-container">
```

```
  <!-- Opportunity cards will be inserted here by JavaScript -->
```

```
  <!-- Example of a card with the bookmark image: -->
```

```
  <div class="opportunity-card">
```

```
    <div class="card-header">
```

```
      <h3>Sample Opportunity Title</h3>
```

```
      
```

```
    </div>
```

```
    <div class="card-details">
```

```
      Internship
```

```
    </div>
```

```
    <button class="card-action-btn save-btn">Save</button>
```

```
  </div>
```

```
</div>
```

```
</section>
```

```
<section class="opportunity-list">
```

```
  <h2>Saved Opportunities</h2>
```

```
  <div id="saved-list" class="card-container">
```

```
    <!-- Saved opportunity cards will be inserted here by JavaScript -->
```

```
        </div>
    </section>
</main>

<script src="app.js"></script>

</body>
</html>
```

2. Styles.css

```
/*
B4 Explanation in Micro:

    Because Flexbox is so good at allocating space along a single axis, it was selected for the two-
    column layout.

    The two primary portions can be positioned side by side on larger screens with this simple and
    efficient method.
*/

/* B2: CSS Custom Properties for a refined color palette */
:root {
    --color-background: #f4f7f9;
    --color-text-base: #333d49;
    --color-text-muted: #8b95a1;
    --color-primary: #3498db;
    --color-accent: #2ecc71;
    --color-danger: #e74c3c;
    --border-color: #dfe3e8;
    --card-background: #ffffff;
    --shadow-color: rgba(0, 0, 0, 0.07);
    --focus-ring-color: #85c1e9;
}

/* --- General Styles and Mobile-First Layout --- */
```

```
body {  
    font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif;  
    margin: 0;  
    background-color: var(--color-background);  
    color: var(--color-text-base);  
    line-height: 1.6;  
}
```

```
/* B5: Base image styles for responsiveness */
```

```
img {  
    max-width: 100%;  
    height: auto;  
    display: block; /* Removes bottom space under the image */  
}
```

```
.site-header {  
    text-align: center;  
    padding: 2rem 1rem;  
    border-bottom: 1px solid var(--border-color);  
    background-color: var(--card-background);  
}
```

```
h1 {  
    margin: 0 0 0.5rem;  
    font-size: 1.8rem;  
    font-weight: 700;  
}
```

```
#opportunity-counter {  
    color: var(--color-text-muted);  
    font-size: 1rem;  
}
```

```
.filter-nav {  
  padding: 1rem;  
  background-color: var(--card-background);  
  border-bottom: 1px solid var(--border-color);  
}
```

```
.filter-controls {  
  display: flex;  
  justify-content: center;  
  gap: 0.75rem;  
}
```

```
.filter-btn {  
  padding: 0.5rem 1.2rem;  
  border: 1px solid var(--border-color);  
  background-color: transparent;  
  color: var(--color-text-muted);  
  border-radius: 999px;  
  cursor: pointer;  
  font-weight: 600;  
  transition: all 0.2s ease-in-out;  
}
```

```
.filter-btn:hover {  
  background-color: #f1f3f5;  
  color: var(--color-text-base);  
}
```

```
.filter-btn.active {  
  background-color: var(--color-primary);  
  border-color: var(--color-primary);  
}
```

```
    color: white;
}

/* B1: Improved visible focus indicator for accessibility */
:focus-visible {
    outline: 3px solid var(--focus-ring-color);
    outline-offset: 2px;
}
input:focus {
    outline: none;
    border-color: var(--color-primary);
    box-shadow: 0 0 0 3px var(--focus-ring-color);
}

.content-area {
    padding: 1.5rem;
    max-width: 1200px;
    margin: 0 auto;
}

.opportunity-list h2 {
    margin-top: 0;
    margin-bottom: 1.5rem;
    padding-bottom: 0.75rem;
    border-bottom: 1px solid var(--border-color);
    color: var(--color-text-base);
    font-size: 1.4rem;
    font-weight: 600;
}

.add-opportunity-form {
```

```
display: flex;
flex-wrap: wrap;
gap: 0.75rem;
margin-bottom: 1.5rem;
}
```

```
.sr-only {
  position: absolute;
  width: 1px;
  height: 1px;
  margin: -1px;
  overflow: hidden;
}
```

```
#opportunity-title-input {
  flex-grow: 1;
  flex-basis: 200px;
  padding: 0.75rem;
  border: 1px solid var(--border-color);
  border-radius: 6px;
  font-size: 1rem;
  transition: all 0.2s ease;
}
```

```
#opportunity-type-input,
#add-opportunity-btn {
  padding: 0.75rem 1rem;
  border: 1px solid var(--border-color);
  border-radius: 6px;
  font-size: 1rem;
  background-color: var(--card-background);
}
```

```
#add-opportunity-btn {  
  background-color: var(--color-accent);  
  color: white;  
  border-color: var(--color-accent);  
  cursor: pointer;  
  font-weight: 600;  
  transition: background-color 0.2s ease;  
}
```

```
#add-opportunity-btn:hover {  
  background-color: #27ae60;  
}
```

```
.error-message {  
  color: var(--color-danger);  
  background-color: #fdeded;  
  border: 1px solid var(--color-danger);  
  border-radius: 6px;  
  padding: 1rem;  
  margin-bottom: 1.5rem;  
  display: none;  
}
```

```
.card-container {  
  display: grid;  
  grid-template-columns: 1fr;  
  gap: 1.5rem;  
}
```

```
.opportunity-card {  
  background-color: var(--card-background);
```



```
border: 1px solid var(--border-color);  
border-radius: 8px;  
padding: 1.5rem;  
box-shadow: 0 4px 6px var(--shadow-color);  
display: flex;  
flex-direction: column;  
}
```

```
.card-header {  
  display: flex;  
  justify-content: space-between;  
  align-items: flex-start;  
  gap: 1rem;  
}
```

```
.bookmark-icon {  
  width: 24px;  
  height: 24px;  
  flex-shrink: 0;  
}
```

```
.opportunity-card h3 {  
  margin: 0;  
  font-size: 1.2rem;  
  color: var(--color-text-base);  
}
```

```
.card-details {  
  color: var(--color-text-muted);  
  font-size: 0.9rem;  
  margin-top: 0.5rem;  
  margin-bottom: 1.5rem;
```

```
    flex-grow: 1;
    text-transform: capitalize;
}
```

```
.card-action-btn {
    width: 100%;
    padding: 0.75rem;
    border-radius: 6px;
    cursor: pointer;
    font-weight: 600;
    font-size: 0.9rem;
    transition: background-color 0.2s ease, color 0.2s ease;
}
```

```
.save-btn {
    background-color: transparent;
    border: 2px solid var(--color-accent);
    color: var(--color-accent);
}
```

```
.save-btn:hover {
    background-color: var(--color-accent);
    color: white;
}
```

```
.remove-btn {
    background-color: transparent;
    border: 2px solid var(--color-danger);
    color: var(--color-danger);
}
```

```
.remove-btn:hover {
    background-color: var(--color-danger);
    color: white;
}
```

```
}
```

```
/* --- B2: Responsive Layout for Screens >= 481px --- */
```

```
@media (min-width: 481px) {
```

```
  .content-area {
```

```
    display: flex;
```

```
    gap: 2rem;
```

```
  }
```

```
  .opportunity-list {
```

```
    flex: 1;
```

```
  }
```

```
}
```

```
/* For wider screens, use a multi-column grid for cards */
```

```
@media (min-width: 900px) {
```

```
  .card-container {
```

```
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
```

```
  }
```

```
}
```

3. App.js

```
/*
```

B4 Micro-explanation:

Event delegation is used to manage clicks on the opportunity cards. Instead of adding a listener to every "Save" button, a single listener is attached to a parent container (`mainContent`).

When a click occurs, `event.target.closest('.card-action-btn')` reliably finds the clicked

button, ensuring the app is memory-efficient and performs well even as cards are added or removed.

```
*/
```

```
// --- Setup and Constants ---

const SID4 = '8480';

const STORAGE_KEY = `opportunityBoard-${SID4}`;

let opportunities = [];

let currentFilter = 'all';


// --- DOM Element References ---

const addButton = document.getElementById('add-opportunity-btn');
const titleInput = document.getElementById('opportunity-title-input');
const typeInput = document.getElementById('opportunity-type-input');
const availableList = document.getElementById('available-list');
const savedList = document.getElementById('saved-list');
const counterSpan = document.getElementById('opportunity-counter');
const mainContent = document.querySelector('.content-area');
const filterContainer = document.querySelector('.filter-controls');
const errorMessageDiv = document.getElementById('error-message');


// --- Main Application Logic ---

function init() {
  loadFromStorage();
  setupEventListeners();
  render();
}

function setupEventListeners() {
  addButton.addEventListener('click', handleAddOpportunity);

  mainContent.addEventListener('click', (event) => {
    const actionButton = event.target.closest('.card-action-btn');
```

```

    if (actionButton) {
      const card = actionButton.closest('.opportunity-card');
      const opportunityId = parseInt(card.dataset.id, 10);
      toggleSavedStatus(opportunityId);
    }
  });

  filterContainer.addEventListener('click', (event) => {
    const filterButton = event.target.closest('.filter-btn');
    if (filterButton) {
      currentFilter = filterButton.dataset.filter;
      render();
    }
  });
}

function handleAddOpportunity() {
  const title = titleInput.value.trim();
  const type = typeInput.value; // type (Internship, Scholarship)

  if (!title) {
    showError("Opportunity title cannot be empty.");
    return;
  }

  hideError();

  const newOpportunity = {
    id: Date.now(),
    title: title,
    category: type,
    saved: false,
  };

```

```
opportunities.push(newOpportunity);  
titleInput.value = '';  
saveAndRender();  
}
```

```
function toggleSavedStatus(id) {  
  const opportunity = opportunities.find(op => op.id === id);  
  if (opportunity) {  
    opportunity.saved = !opportunity.saved;  
    saveAndRender();  
  }  
}
```

```
function render() {  
  const filteredOpportunities = opportunities.filter(op => {  
    if (currentFilter === 'all') return true;  
    return op.category === currentFilter;  
  });
```

```
  availableList.innerHTML = '';  
  savedList.innerHTML = '';
```

```
  for (const op of filteredOpportunities) {  
    const cardElement = createOpportunityCard(op);  
    if (op.saved) {  
      savedList.appendChild(cardElement);  
    } else {  
      availableList.appendChild(cardElement);  
    }  
  }  
}
```

```

    updateCounter();
    updateActiveFilter();
}

function createOpportunityCard(opportunity) {
    const card = document.createElement('div');
    card.className = 'opportunity-card';
    card.dataset.id = opportunity.id;

    const buttonText = opportunity.saved ? 'Remove' : 'Save';
    const buttonClass = opportunity.saved ? 'card-action-btn remove-btn' : 'card-action-btn save-
btn';

    card.innerHTML = `
        <h3>${escapeHTML(opportunity.title)}</h3>
        <p class="card-details">Category: ${escapeHTML(opportunity.category)}</p>
        <button class="${buttonClass}">${buttonText}</button>
    `;
    return card;
}

function updateCounter() {
    const savedCount = opportunities.filter(op => op.saved).length;
    counterSpan.textContent = `${opportunities.length} opportunities (${savedCount} saved)`;
}

function updateActiveFilter() {
    document.querySelectorAll('.filter-btn').forEach(btn => {
        btn.classList.toggle('active', btn.dataset.filter === currentFilter);
    });
}

// --- Persistence ---

```

```
function saveToStorage() {  
    localStorage.setItem(STORAGE_KEY, JSON.stringify(opportunities));  
}
```

```
function loadFromStorage() {  
    const storedData = localStorage.getItem(STORAGE_KEY);  
    if (storedData) {  
        opportunities = JSON.parse(storedData);  
    }  
}
```

```
function saveAndRender() {  
    saveToStorage();  
    render();  
}
```

```
// --- Utility Functions ---
```

```
function showError(message) {  
    errorMessageDiv.textContent = message;  
    errorMessageDiv.style.display = 'block';  
}
```

```
function hideError() {  
    errorMessageDiv.style.display = 'none';  
}
```

```
function escapeHTML(str) {  
    const p = document.createElement('p');  
    p.appendChild(document.createTextNode(str));  
    return p.innerHTML;
```

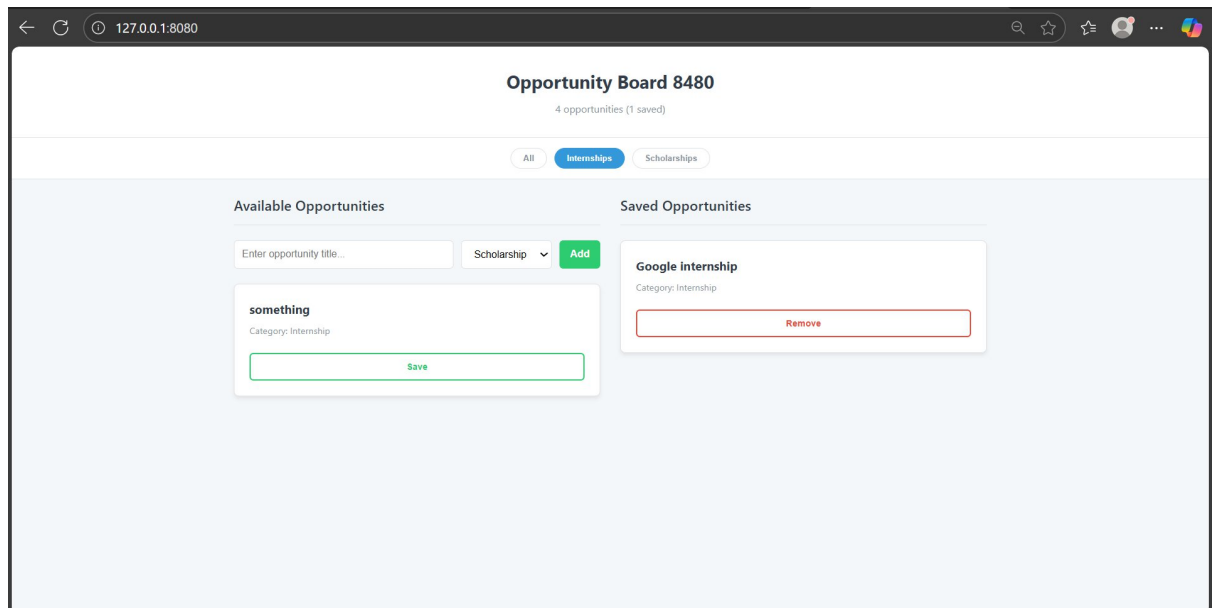


```
}
```

```
// --- Entry Point ---
```

```
document.addEventListener('DOMContentLoaded', init);
```

4. screenshots



5. Design and Analysis

a. Analysis

An entire interactive "Opportunity Board" web application is produced by this code. In order to display "Available" and "Saved" chances, the HTML file sets up the page's basic layout, which includes a header, filter buttons, and two main sections. All of the visual styling is provided by the CSS file, giving the page a crisp, contemporary appearance. Its adaptable design adjusts the layout to fit many screen sizes, including desktops and mobile phones. The page comes to life thanks to the JavaScript file. Users can restrict the view to only see internships or scholarships, add new chances, and click a button to transfer them between the "Available" and "Saved" lists. It instantly stores all of the opportunities to the browser's local storage to guarantee that user data is preserved.

b. Design

This CSS creates a clean, modern, and responsive design for the Opportunity Board. It uses custom properties (variables) for a consistent color scheme and spacing, and employs Flexbox and Grid to organize the layout so it looks great on any device, from mobile phones to desktops. The styling also includes subtle shadows, transitions, and clear focus indicators to enhance user experience.