```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Select Your Avatar</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css">
    <style>
        body {
            background-color: #f8f9fa;
            font-family: Arial, sans-serif;
        }
        .container {
            margin-top: 50px;
            max-width: 900px;
        }
        .card {
            border: none;
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            overflow: hidden;
            transition: transform 0.3s ease, box-shadow 0.3s ease;
        }
        .card:hover {
            transform: translateY(-5px);
            box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
        }
        .card img {
            width: 100%;
            height: 285px;
            object-fit: cover;
        }
        .card-body {
            padding: 15px;
        }
        .card-title {
            font-size: 1rem;
            font-weight: bold;
            margin-bottom: 0px;
```

```css
    text-align: center;
}
.skills {
    font-size: 0.9rem;
    color: #6c757d;
}
.action-buttons {
    text-align: center;
    margin-top: 20px;
}
.action-buttons button {
    margin: 22px 10px;
    margin-top: -11px;
}
#loading {
    text-align: center;
    margin-top: 50px;
    font-size: 1.5rem;
}
#emptyState {
    display: none;
    text-align: center;
    margin-top: 50px;
}
/* Add styles for the modal */
.modal {
    display: none;
    position: fixed;
    z-index: 1000;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
}

.modal-content {
    background-color: #fefefe;
    margin: 15% auto;
    padding: 20px;
```

```
  border: none;
  border-radius: 10px;
  width: 80%;
  max-width: 600px;
  position: relative;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.modal-header {
  margin-bottom: 15px;
}

.modal-title {
  margin: 0;
  color: #333;
}

.modal-body {
  color: #666;
  line-height: 1.6;
}

.close {
  position: absolute;
  right: 20px;
  top: 15px;
  font-size: 28px;
  font-weight: bold;
  color: #aaa;
  cursor: pointer;
}

.close:hover {
  color: #333;
}

.title-section {
  padding: 2rem 0;
  margin-bottom: 3rem;
}
```

```
        .title-section h1 {
            color: #2c3e50;
            font-weight: bold;
            font-size:1.5rem;
        }

        .title-section .lead {
            font-size: 1rem;
            color: #6c757d;
        }

        /* Adjust container top margin since we now have a title section */
        .container {
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <!-- Add modal HTML -->
    <div id="avatarModal" class="modal">
        <div class="modal-content">
            <span class="close">&times;</span>
            <div class="modal-header">
                <h4 class="modal-title"></h4>
            </div>
            <div class="modal-body"></div>
        </div>
    </div>

    <div class="container">
        <!-- Add title section -->
        <div class="title-section text-center mb-0">
            <h1 class="mb-3">Select Your Archetypes</h1>
            <p class="lead text-muted">Choose the archetypes that best represent you</p>
        </div>

        <div id="loading">Loading avatars...</div>
        <div id="emptyState">
            <h4>No avatars available!</h4>
```

```
      <p>Please try again later.</p>
    </div>
    <div class="row" id="avatarGrid" style="display: none;">
      <!-- Avatar cards will be added dynamically -->
    </div>
    <div class="action-buttons">
      <button class="btn btn-primary" id="saveSelection" style="display: none;">Save
Selection</button>
      <button class="btn btn-secondary" id="clearSelection" style="display:
none;">Clear Selection</button>
    </div>
  </div>

  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      const avatarGrid = $('#avatarGrid');
      const loading = $('#loading');
      const emptyState = $('#emptyState');
      const saveSelectionButton = $('#saveSelection');
      const clearSelectionButton = $('#clearSelection');
      const username = localStorage.getItem('username');
      const modal = $('#avatarModal');
      const modalTitle = $('.modal-title');
      const modalBody = $('.modal-body');
      const closeBtn = $('.close');
      let allAvatarsData = [];

      if (!username) {
        alert('No user logged in! Redirecting...');
        window.location.href = 'login.html';
        return;
      }

      // Fetch avatars and user data from backend
      fetch('/app-data')
        .then(response => {
          if (!response.ok) {
            throw new Error('Failed to fetch avatars.');
          }
          return response.json();
        })
        .then(data => {
          loading.hide();
          allAvatarsData = data.allAvatars;

          if (data.allAvatars && data.allAvatars.length > 0) {
            avatarGrid.show();
            saveSelectionButton.show();
            clearSelectionButton.show();

            // Populate avatar grid
            data.allAvatars.forEach((avatar, index) => {
              const backgroundImage = avatar.backgroundImage ||
`${avatar.name.replace(/\s+/g, '_')}_color.png`;
              const card = `
                <div class="col-md-4 mb-4">
                  <div class="card">
                    <img src="img/Archtype Image/${backgroundImage}"
                         alt="${avatar.name}"
                         class="avatar-image"
                         data-avatar-name="${avatar.name}"

onerror="this.src='https://via.placeholder.com/300x200?text=${encodeURIComponent(
avatar.name)}'">
                    <div class="card-body">
                      <h5 class="card-title">
                        <input type="checkbox" id="avatar-${index}"
value="${avatar.name}">
                        <label for="avatar-${index}">${avatar.name}</label>
                      </h5>
                    </div>
                  </div>
                </div>
              `;
              avatarGrid.append(card);
            });

            // Pre-check previously selected avatars
            const currentUser = data.users.find(user => user.name === username);
            if (currentUser && currentUser.selectedAvatars) {
                currentUser.selectedAvatars.forEach(selectedAvatar => {
                    $(`input[value="${selectedAvatar}"]`).prop('checked', true);
                });
            }
        } else {
            emptyState.show();
        }
    })
    .catch(error => {
        console.error('Error fetching avatars:', error);
    });

// Modify only the click handler for avatar images
$(document).on('click', '.avatar-image', function() {
    const avatarName = $(this).data('avatar-name');
    const avatar = allAvatarsData.find(a => a.name === avatarName);

    if (avatar) {
        modalTitle.text(avatar.name);
        const description = avatar.description || 'No description available.';
        modalBody.html(`
            <p>${description}</p>
        `);
        modal.css('display', 'block');
    }
});

// Close modal when clicking the close button or outside the modal
closeBtn.click(() => modal.css('display', 'none'));
$(window).click(function(event) {
    if ($(event.target).is(modal)) {
        modal.css('display', 'none');
    }
});

// Save Selection
saveSelectionButton.on('click', function () {
    const selectedAvatars = Array.from(
          avatarGrid.find('.card-title input[type="checkbox"]:checked')
        ).map(input => input.value);

        if (selectedAvatars.length > 0) {
          // Fetch existing data and update the user's selectedAvatars
          fetch('/app-data')
            .then(response => response.json())
            .then(data => {
              const users = data.users || [];
              const currentUserIndex = users.findIndex(user => user.name ===
username);

              if (currentUserIndex !== -1) {
                users[currentUserIndex].selectedAvatars = selectedAvatars;

                // Prepare the updated JSON
                const updatedData = {
                  ...data,
                  users: users
                };

                // Send the updated data back to the backend
                return fetch('/app-data', {
                  method: 'POST',
                  headers: {
                    'Content-Type': 'application/json'
                  },
                  body: JSON.stringify(updatedData)
                });
              } else {
                console.error('User not found.');
              }
            })
            .then(() => {
              // Add redirect to select-skills.html after successful save
              window.location.href = 'select-skills.html';
            })
            .catch(error => console.error('Error saving selected avatars:', error));
        }
      });
```

```
            // Clear Selection
            clearSelectionButton.on('click', function () {
                avatarGrid.find('.card-title input[type="checkbox"]').prop('checked', false);
            });
        });
    </script>
</body>
</html>
```