5 Quick jQuery Classroom Projects — Full Code Pack

Each section contains a complete, copy-paste-ready single HTML file (with jQuery via CDN). You can distribute this DOCX or export to PDF yourself. If offline, download 'jquery-3.7.1.min.js' and update the <script src> path.

1) To-Do List with LocalStorage (CRUD + events)

What students learn: Selectors, event delegation, DOM building, toggling, LocalStorage persistence.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>jQuery To-Do (LocalStorage)</title>
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <style>
    body{font:16px/1.4 system-ui, sans-serif; max-width:720px; margin:40px
auto; padding:0 16px}
    h1{margin:0 0 12px}
    form{display:flex; gap:8px; margin:10px 0 20px}
    input[type="text"]{flex:1; padding:10px}
    button{padding:10px 14px; cursor:pointer}
    ul{list-style:none; padding:0; margin:0}
    li{display:flex; align-items:center; justify-content:space-between;
gap:12px; padding:10px 12px; border:1px solid #ddd; border-radius:10px;
margin:8px 0}
    li.done .txt{opacity:.6; text-decoration:line-through}
    .actions{display:flex; gap:6px}
    .empty{color:#666; font-style:italic}
  </style>
</head>
<body>
  <h1>To-Do</h1>
  <form id="addForm">
    <input id="newItem" type="text" placeholder="Add a task…" required />
    <button>Add</button>
  </form>
  <ul id="list"></ul>
  <p class="empty" id="emptyMsg">No tasks yet — add one!</p>

  <script>
    const KEY = 'jq_todos_v1';

    function save(){
      const data = $('#list li').map(function(){
        return { text: $(this).find('.txt').text(), done:
$(this).hasClass('done') };
      }).get();
      localStorage.setItem(KEY, JSON.stringify(data));
      toggleEmpty();
    }

    function toggleEmpty(){
      $('#emptyMsg').toggle($('#list li').length === 0);
    }

    function itemTemplate(text, done){
      return $(
        `<li class="${done ? 'done' : ''}">
           <span class="txt"></span>
           <div class="actions">
             <button class="toggle" type="button">✔</button>
             <button class="remove" type="button">✖</button>
           </div>
         </li>`
      ).find('.txt').text(text).end();
    }

    $(function(){
      try{
        const saved = JSON.parse(localStorage.getItem(KEY) || '[]');
        saved.forEach(it => $('#list').append(itemTemplate(it.text, it.done)));
      }catch(e){}
      toggleEmpty();

      $('#addForm').on('submit', function(e){
        e.preventDefault();
        const val = $('#newItem').val().trim();
        if(!val) return;
        $('#list').append(itemTemplate(val,false));
        $('#newItem').val('').focus();
        save();
      });

      $('#list').on('click', 'button.toggle', function(){
        $(this).closest('li').toggleClass('done');
        save();
      });

      $('#list').on('click', 'button.remove', function(){
        $(this).closest('li').slideUp(120, function(){ $(this).remove();
save(); });
      });
    });
  </script>
</body>
</html>
```

Demo tips: Add, toggle, delete, then refresh page to show persistence.

2) Image Gallery + Lightbox (effects + keyboard)

What students learn: Click handlers, data-* attributes, fade/slide effects, keyboard navigation, simple state.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>jQuery Lightbox Gallery</title>
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <style>
    body{font:16px/1.4 system-ui, sans-serif; max-width:960px; margin:40px
auto; padding:0 16px}
    .grid{display:grid; grid-template-columns:repeat(auto-
fill,minmax(160px,1fr)); gap:12px}
    .grid img{width:100%; height:120px; object-fit:cover; border-radius:12px;
cursor:pointer;}
    .lb{position:fixed; inset:0; background:rgba(0,0,0,.85); display:none;
align-items:center; justify-content:center}
    .lb img{max-width:88vw; max-height:80vh; border-radius:12px}
    .lb .nav{position:fixed; inset:0; display:flex; align-items:center;
justify-content:space-between; padding:0 24px; color:#fff; font-size:28px;
user-select:none}
    .lb .close{position:fixed; top:14px; right:20px; font-size:28px;
color:#fff; cursor:pointer}
  </style>
</head>
<body>
  <h1>Gallery</h1>
  <div class="grid" id="grid"></div>

  <div class="lb" id="lightbox" aria-hidden="true">
    <span class="close" title="Close">✖</span>
    <div class="nav">
      <span class="prev" title="Previous">❬</span>
      <span class="next" title="Next">❭</span>
    </div>
    <img id="lbImg" alt="Preview"/>
  </div>

  <script>
    const imgs = Array.from({length:9},
(_,i)=>`https://picsum.photos/seed/jq${i+1}/800/600`);
    let idx = 0;

    function openAt(i){
      idx = (i+imgs.length)%imgs.length;
      $('#lbImg').attr('src', imgs[idx]);
      $('#lightbox').fadeIn(120).attr('aria-hidden','false');
    }

    $(function(){
      imgs.forEach((src,i)=>$('#grid').append(`<img src="${src}" data-i="${i}"
alt="Thumb ${i+1}">`));

      $('#grid').on('click','img', function(){ openAt(+$(this).data('i')); });

      $('.next').on('click', ()=>openAt(idx+1));
      $('.prev').on('click', ()=>openAt(idx-1));

      $('.close, #lightbox').on('click', function(e){
        if(e.target !== this && !$(e.target).is('.close')) return;
        $('#lightbox').fadeOut(120).attr('aria-hidden','true');
      });

      $(document).on('keydown', function(e){
        if($('#lightbox').is(':hidden')) return;
        if(e.key==='Escape') $('.close').click();
        if(e.key==='ArrowRight') $('.next').click();
        if(e.key==='ArrowLeft') $('.prev').click();
      });
    });
  </script>
</body>
</html>
```

Demo tips: Open a thumbnail, use ←/→ keys, click backdrop to close.

3) Live Search Filter (keyup + highlight)

What students learn: Keyup events, filtering with selectors, small debounce, result count, simple highlighting.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>jQuery Live Search</title>
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <style>
    body{font:16px/1.4 system-ui, sans-serif; max-width:720px; margin:40px
auto; padding:0 16px}
    input{width:100%; padding:10px; border:1px solid #ddd; border-radius:10px;}
    ul{list-style:none; padding:0}
    li{padding:8px 10px; border-bottom:1px solid #eee}
    mark{background:#ffec99; padding:0 3px}
    .muted{color:#666}
  </style>
</head>
<body>
  <h1>Live Search</h1>
  <input id="q" type="text" placeholder="Type to filter… (e.g., an, ar, li)"/>
  <p class="muted"><span id="count"></span></p>
  <ul id="list"></ul>
  <script>
    const items =
['Apple','Banana','Cherry','Date','Elderberry','Fig','Grape','Honeydew','Kiwi',
'Lemon','Mango','Nectarine','Orange','Papaya','Quince','Raspberry','Strawberry'
,'Tangerine','Ugli fruit','Watermelon'];

    function render(list, term=''){
      const $ul = $('#list').empty();
      const re = term ? new
RegExp('('+term.replace(/[.*+?^${}()|[\\]\\\\]/g,'\\$&')+')','i') : null;
      list.forEach(txt => {
        const html = re ? txt.replace(re,'<mark>$1</mark>') : txt;
        $('<li>').html(html).appendTo($ul);
      });
      $('#count').text(`${list.length} result${list.length===1?'':'s'}`);
    }

    $(function(){
      render(items);
      let t;
      $('#q').on('keyup', function(){
        clearTimeout(t);
        const val = this.value.trim().toLowerCase();
        t = setTimeout(()=>{
          const filtered = items.filter(x => x.toLowerCase().includes(val));
          render(filtered, val);
        }, 120);
      });
    });
  </script>
</body>
</html>
```

Demo tips: Type fragments like “an”, “ar”, “li”; show highlight & count.

4) Friendly Form Validation + Character Counter

What students learn: Intercept submit, regex checks, inline feedback, class toggling, basic UX.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>jQuery Form Validation</title>
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <style>
    body{font:16px/1.4 system-ui, sans-serif; max-width:720px; margin:40px
auto; padding:0 16px}
    label{display:block; margin:14px 0 6px}
    input, textarea{width:100%; padding:10px; border:1px solid #ddd; border-
radius:10px}
    small{display:block; color:#666}
    .error{border-color:#e8590c; background:#fff5f0}
    .ok{border-color:#2f9e44; background:#f4fff7}
    .msg{font-size:14px; color:#e8590c; height:18px}
    .msg.ok{color:#2f9e44}
    button{margin-top:12px; padding:10px 14px}
  </style>
</head>
<body>
  <h1>Contact Us</h1>
  <form id="f">
    <label>Email</label>
    <input id="email" type="email" placeholder="you@example.com" />
    <div class="msg" id="mEmail"></div>

    <label>Password</label>
    <input id="pw" type="password" placeholder="At least 8 chars" />
    <div class="msg" id="mPw"></div>

    <label>Message <small>(max 140 chars)</small></label>
    <textarea id="msg" rows="4" maxlength="140"></textarea>
    <small><span id="left">140</span> characters left</small>

    <button>Send</button>
  </form>

  <script>
    function setState($el, ok, $msg, text){
      $el.toggleClass('ok', ok).toggleClass('error', !ok);
      $msg.text(text).toggleClass('ok', ok);
    }

    $(function(){
      const reEmail = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

      $('#email').on('input', function(){
        const ok = reEmail.test(this.value.trim());
        setState($(this), ok, $('#mEmail'), ok ? 'Looks good ✓' : 'Please enter
a valid email');
      });

      $('#pw').on('input', function(){
        const v = this.value;
        const ok = v.length >= 8 && /[A-Z]/.test(v) && /[0-9]/.test(v);
        setState($(this), ok, $('#mPw'), ok ? 'Strong enough ✓' : 'Min 8 chars,
include A-Z and 0-9');
      });

      $('#msg').on('input', function(){
        const left = 140 - this.value.length;
        $('#left').text(left);
      });

      $('#f').on('submit', function(e){
        e.preventDefault();
        $('#email').trigger('input');
        $('#pw').trigger('input');
        const ok = $('#email').hasClass('ok') && $('#pw').hasClass('ok');
        if(ok){
          alert('Submitted! (demo only)');
          this.reset();
          $('.msg').text('').removeClass('ok');
          $('input, textarea').removeClass('ok error');
          $('#left').text('140');
        } else {
          alert('Please fix the errors.');
        }
      });
    });
  </script>
</body>
</html>
```

Demo tips: Type invalid→valid values; show real-time validation and counter.

5) FAQ Accordion with Smooth Toggle + Deep-Linking

What students learn: slideToggle, managing active state, closing siblings, URL hash deep-linking.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>jQuery FAQ Accordion</title>
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <style>
    body{font:16px/1.4 system-ui, sans-serif; max-width:720px; margin:40px
auto; padding:0 16px}
    .item{border:1px solid #ddd; border-radius:10px; margin:10px 0;
overflow:hidden}
    .q{padding:12px 14px; background:#f8f9fa; cursor:pointer; font-weight:600}
    .a{display:none; padding:14px}
    .item.active .q{background:#e7f5ff}
  </style>
</head>
<body>
  <h1>FAQ</h1>
  <div class="acc" id="faq">
    <div class="item" id="q1">
      <div class="q">What is jQuery?</div>
      <div class="a">A small JavaScript library that makes DOM, events, and
effects easier across browsers.</div>
    </div>
    <div class="item" id="q2">
      <div class="q">Should we still learn it?</div>
      <div class="a">Yes — many legacy projects use it, and it teaches
event-driven UI patterns.</div>
```

```
    </div>
    <div class="item" id="q3">
      <div class="q">How big is jQuery?</div>
      <div class="a">About ~30 KB minified + gzipped for core features
(v3.7.x).</div>
    </div>
  </div>

  <script>
    function openItem($it, push=true){
      if(!$it.length) return;
      $it.addClass('active').children('.a').stop(true,true).slideDown(140);

$it.siblings('.item').removeClass('active').children('.a').stop(true,true).slid
eUp(140);
      if(push) location.hash = $it.attr('id');
    }

    $(function(){
      $('#faq').on('click', '.q', function(){
        const $it = $(this).closest('.item');
        if($it.hasClass('active')){
          $it.removeClass('active').children('.a').slideUp(140);
          history.pushState('', document.title, window.location.pathname +
window.location.search);
        } else {
          openItem($it);
        }
      });

      const id = location.hash.slice(1);
      openItem($('#'+id), false);

      $(window).on('hashchange', function(){
        const id = location.hash.slice(1);
        if(id){ openItem($('#'+id), false); }
      });
    });
  </script>
</body>
</html>
```

Demo tips: Click questions; copy a #q2 link, reload to show deep-linking.