# MIS 2201 Exam 2
## Topic Review Checklist

40 Questions — 150 Points — 65 Minutes
All questions equal weight (3.75 points each)
**Materials Allowed:** One page (single-sided) cheat sheet

**Detailed Content Reference**

For comprehensive explanations, code examples, and interactive practice:
**Building Websites Lecture:**
Click here for full lecture content
This quick reference guide covers HTML/CSS topics from the lecture above. Project Management and E-commerce content is from your textbook.

## HTML Basics (8 Questions)

*HTML structure, semantic elements, nesting rules*

**Document Structure:**

- `<!DOCTYPE html>` → tells browser: "Use HTML5!"
- `<head>` = invisible info (title, CSS, meta)
- `<body>` = visible content (what users see)

**Semantic HTML (Use Meaning, Not Just Style):**

- WHY? Accessibility + SEO + meaning
- **Memory trick:** Think of a newspaper layout
- `<header>` = masthead/top
- `<nav>` = navigation menu
- `<main>` = main article area
- `<article>` = individual story
- `<section>` = story sections
- `<aside>` = sidebar/ads
- `<footer>` = bottom info

**Nesting Rules (CRITICAL):**

- INVALID: Inline cannot contain block
- INVALID: `<p>` cannot contain `<div>`
- INVALID: `<p>` cannot contain `<p>`
- INVALID: `<a>` cannot contain `<a>`
- VALID: Block can contain inline/block

**Block vs Inline (Know the Difference!):**

- **Block:** New line, full width, stackable
- Examples: div, p, h1-h6, ul, section, header
- **Inline:** Flows with text, only what's needed
- Examples: span, a, strong, em, img, code
- Quiz yourself: Is `<nav>` block or inline? (Answer: Block!)

## HTML Advanced (5 Questions)

*Forms, accessibility, media embedding*

**Accessible Forms (Screen Readers Need This!):**

- `<label for="email">` + `<input id="email">`
- Connection = `for` attribute matches `id`
- `<fieldset>` + `<legend>` = group related fields
- NEVER: placeholder as label replacement

**Input Types (Know the Differences):**

- `type="text"` → any text
- `type="radio"` → **ONE** choice (like MCQ)
- Same `name` groups radio buttons together
- `type="checkbox"` → **MULTIPLE** choices
- Exam tip: Radio = exclusive, Checkbox = inclusive

**Media Embedding (Images & Video):**

- `<img src="pic.jpg" alt="description">`
- `alt` = accessibility & broken image fallback
- `width/height` = prevent layout jump
- `<video controls>` + `<source>` tags
- `<audio>` for sound files
- `<picture>` + `srcset` = responsive images
- Remember: Always include `alt` text!

**Responsive Video (The Padding Trick):**

- Problem: Fixed iframe width breaks on mobile
- Solution: Wrapper div with padding-bottom
- 16:9 ratio = 56.25% padding-bottom
- Math: $9 \div 16 \times 100 = 56.25\%$
- **Quick formula:** height ÷ width × 100 = %

## CSS Basics (7 Questions)

*Selectors, specificity, box model, units*

**CSS Syntax (Basic Pattern):**

- `selector { property: value; }`
- Element: `p {}` → all paragraphs
- Class: `.button {}` → elements with class="button"
- ID: `#header {}` → element with id="header"
- Descendant: `nav a {}` → links inside nav

**Specificity (MOST TESTED CONCEPT!):**

- **Formula:** (IDs - Classes - Elements)
- `p` = 0-0-1 (1 element)
- `.button` = 0-1-0 (1 class)
- `#header` = 1-0-0 (1 ID)
- `div.container` = 0-1-1 (1 class + 1 element)
- `#header .nav li` = 1-1-1 (1 ID + 1 class + 1 element)
- Rule: Higher number wins!
- Tie? Last rule in CSS file wins
- **Memory:** IDs are most specific, elements least

**Box Model (Key Concept):**

- **Order (inside out):** Content → Padding → Border → Margin
- **Visual:** [Margin [Border [Padding [Content]]]]
- **Total Width =** margin + border + padding + width + padding + border + margin
- Margin Collapse: Vertical margins merge (use larger value)
- Horizontal margins DON'T collapse (they add up)
- **Exam tip:** Always include ALL parts in width calculations!

**CSS Units (When to Use Each):**

- `px` - fixed size, borders, shadows
- `%` - responsive widths, relative to parent
- `em` - spacing relative to current font size
- `rem` - consistent sizing based on root (best for fonts!)
- `vh/vw` - full-screen sections (100vh = full height)
- **Remember:** `rem` ¿ `em` for scalability

**CSS Inheritance (Some Pass Down, Some Don't):**

- DO inherit: color, font-*, text-align, line-height
- DON'T inherit: margin, padding, border, width, height, background
- **Exception:** Links (`<a>`) don't inherit color (need explicit styling)

## CSS Layout (5 Questions)

*Flexbox, Grid, responsive design*

**Flexbox (1D Layout - Row OR Column):**

- `display: flex` on parent container
- `flex-direction: row` (default) or `column`
- `justify-content` = main axis (horizontal in row)
- Values: `flex-start, center, space-between, space-around`
- `align-items` = cross axis (vertical in row)
- `gap: 20px` = spacing between items (easiest way!)
- **Use when:** Nav bars, centering, single row/column
- Perfect for: Aligning items in one direction

**CSS Grid (2D Layout - Rows AND Columns):**

- `display: grid` on parent container
- `grid-template-columns: 1fr 1fr 1fr` (3 equal columns)
- `grid-template-rows: auto auto` (2 rows)
- `grid-template-areas: "header header" "main aside";`
- Repeat name = cell spans multiple areas
- Dot (.) = empty cell placeholder
- **Use when:** Full page layouts, cards, complex designs
- Perfect for: Two-dimensional layouts

**Flexbox vs Grid (Know When to Use):**

- **Flexbox:** 1D, content flows, flexible sizing
- **Grid:** 2D, fixed structure, precise control
- Quiz: Product cards in rows? Flexbox or Grid!
- Quiz: Page with header/main/sidebar? Grid!

**Responsive Design (Mobile First!):**

- `@media (min-width: 768px) { /* styles */ }`
- `min-width: 768px` = "if screen is AT LEAST 768px wide"
- `max-width: 767px` = "if screen is AT MOST 767px wide"
- **Common breakpoints:**
- 768px = tablet/iPad portrait
- 1024px = desktop/iPad landscape
- Mobile-first approach: Base styles = mobile, `min-width` for larger
- **Exam tip:** Read "min-width" as "minimum width required"

## Project Management Phases (3 Questions)

*SMART criteria, PM phases*

**S.M.A.R.T. Goals (Every Goal Should Be):**

- **S**pecific - crystal clear objective (Who? What? Where?)
- **M**easurable - can track progress with numbers/metrics
- **A**ttainable - possible to achieve (not fantasy)
- **R**ealistic - feasible with available resources
- **T**ime-bound - has deadline/timeframe
- Example: "Increase sales by 20% by Q4" = SMART!
- Bad: "Improve business" = Not SMART

**Four PM Phases (In Order):**

- **1. Initiation:** "Should we do this?"
- → Feasibility, business case, identify stakeholders, define scope
- **2. Planning:** "How will we do this?"
- → Break into tasks, Gantt chart, budget, assign resources
- **3. Execution:** "Let's do this!"
- → Implement plan, monitor progress, track milestones, adjust
- **4. Closure:** "Did we do it right?"
- → Review performance, lessons learned, close contracts, celebrate!
- **Memory trick:** I.P.E.C. = "I Plan, Execute, Close"

## Project Management Tools (4 Questions)

*Flowcharts, Gantt, PERT charts*

**Flowchart Symbols (Basic Process Visualization):**

- **Oval** = Start/End points
- **Rectangle** = Process/activity/action
- **Diamond** = Decision (Yes/No branch)
- **Arrow** = Flow direction
- Best for: Simple workflows everyone can understand
- Non-technical stakeholders love these!

**Gantt Charts (Timeline Visualization):**

- **Y-axis** = Task names
- **X-axis** = Time (days/weeks/months)
- Horizontal bars show task duration
- Shows which tasks run concurrently (parallel bars)
- Shows dependencies (arrows between bars)
- Best for: Planning phase, seeing "when" things happen
- **Example:** Microsoft Project uses Gantt charts

**PERT Charts (Network Diagrams):**

- Network of circles (events) and arrows (tasks)
- Shows complex task interdependencies
- Identifies **critical path** (longest sequence)
- Helps with resource allocation decisions
- Best for: Complex projects, optimizing schedule
- **Key feature:** Finds bottlenecks and critical tasks

**Tool Selection (Exam Question Type):**

- Need simple visual for presentation? → **Flowchart**
- Need to show timeline/schedule? → **Gantt Chart**
- Need to find critical path? → **PERT Chart**
- Remember: Match tool to project need!

## PM Advanced Concepts (3 Questions)

*Risk management, critical path*

**Risk Management (4 Strategies):**

- **1. Avoidance:** Eliminate the risk completely
- Example: Change supplier to avoid late deliveries
- **2. Transference:** Shift risk to someone else
- Example: Buy insurance, outsource risky work
- **3. Mitigation:** Reduce the risk's impact
- Example: Keep backup equipment, train extra staff
- **4. Acceptance:** Acknowledge and live with it
- Example: Low-probability, low-impact risks
- **Memory:** A.T.M.A. = "Avoid, Transfer, Mitigate, Accept"

**Critical Path Method (CPM - Key Concept!):**

- **Critical path** = LONGEST path through project
- Determines MINIMUM project duration
- Delay on critical path = entire project delays
- Non-critical tasks have "slack time" (can delay without impact)
- **Example:** A→B→C (10 days) vs A→D→C (8 days)
- Critical path = A→B→C (longer!)

**Project Crashing (Speed Up!):**

- Shortening critical path activities
- Add more resources (people, money, equipment)
- Cost increases!
- Used when deadline is non-negotiable
- **Exam tip:** Only crash critical path tasks!

**Resource Leveling (Smooth It Out):**

- Balance workload when resources are limited
- Adjust non-critical tasks to avoid resource conflicts
- Example: 2 tasks need same person → stagger them

## E-commerce Basics (3 Questions)

*Transaction types, m-commerce*

**Transaction Types (Direction Matters!):**

- **B2C:** Business → Consumer
- Examples: Amazon, Walmart.com, Netflix
- **B2B:** Business → Business
- Examples: Office supplies wholesaler, industrial equipment
- **B2G:** Business → Government
- Examples: RFP bids, government contracts
- **C2C:** Consumer → Consumer
- Examples: eBay, Craigslist, Facebook Marketplace
- **C2B:** Consumer → Business
- Examples: Fiverr (freelancer to company), Upwork
- **C2A:** Consumer → Admin/Government
- Examples: Pay taxes online, renew license, permits
- Quiz: Freelancer on Fiverr? C2B!

**M-commerce (Mobile E-commerce):**

- E-commerce transactions via mobile devices
- NOT separate from e-commerce - it's a **subset**!
- Examples: Mobile banking, app purchases, Apple Pay, mobile shopping
- **Key feature:** Location-based services

**Key E-commerce Concepts:**

- **Ubiquity:** Available everywhere, anytime, any device
- **Personalization:** Marketing tailored TO you (company does it)
- **Customization:** You adjust product/service to YOUR preferences
- Don't confuse! Personalization = they do it, Customization = you do it

## AI in E-commerce (2 Questions)

*AI applications*

**AI Applications (How AI Helps E-commerce):**

- **Customer Support:** Chatbots answer questions 24/7
- Multilingual support, instant responses, reduces costs
- **Dynamic Pricing:** Adjust prices in real-time
- Based on: demand, competitor prices, customer behavior
- Example: Airline/hotel prices changing constantly
- **Inventory Management:** Predict what to stock
- Automated reordering, warehouse robot optimization
- **Fraud Detection:** Catch suspicious transactions
- Pattern recognition, real-time monitoring, reduce chargebacks
- **Predictive Analytics:** Forecast trends
- Predict demand, personalize recommendations, anticipate needs
- Example: "Customers who bought X also bought Y"

### Cheat Sheet Suggestions

**What to Include on Your Cheat Sheet (One Side, One Page):**

- CSS specificity formula with examples
- Box model diagram + width calculation
- Flexbox properties (`justify-content`, `align-items` values)
- Grid template-areas example
- Media query syntax
- PM phases (Initiation, Planning, Execution, Closure)
- Flowchart symbols diagram
- Risk strategies (Avoidance, Transference, Mitigation, Acceptance)
- Transaction types (B2C, B2B, B2G, C2C, C2B)
- Common CSS units and when to use

## Common Mistakes to Avoid

**HTML/CSS:**

- Thinking inline can contain block elements
- Not accounting for all box model parts in width
- Confusing Flexbox (1D) with Grid (2D)
- Forgetting margin collapse (vertical only)

**Project Management:**

- Mixing up PM phase activities
- Not knowing when to use which tool
- Thinking all paths are critical (only longest)

**E-commerce:**

- Confusing B2C with C2B (direction matters!)
- Thinking m-commerce is separate (it's a subset)
- Mixing up personalization vs customization

## Exam Strategy

- **Time:** 65 min ÷ 40 questions = 1.6 min/question
- **First pass (25-30 min):** Answer easy questions
- **Second pass (20-25 min):** Tackle harder questions
- **Final pass (10-15 min):** Review and check
- Use your cheat sheet strategically
- Read each question carefully
- Watch for code-based questions (8-10 of them)
- Don't spend too long on any one question

**Good luck on your exam!**

*Remember your cheat sheet!*