

```

**ABSOLUTE EXCEPTIONS - NO CODE GENERATION:**
- **Time Anchor:** bullets - NEVER EVER generate any code, programming solutions, or technical implementations
- **Personal Introduction:** bullets - NEVER EVER generate any code, programming solutions, or technical implementations
- **FOR PERSONAL INTRODUCTION BULLETS:** Generate questions FROM the perspective of the person introducing themselves
The person in the bullet is the USER asking the questions
These are contextual/personal details, NOT technical coding scenarios

**STOP AND VERIFY: If the bullet contains "Time Anchor:" or "Personal Introduction:" in the title, you MUST
NOT generate ANY code blocks, programming solutions, scripts, or technical implementations. Period.**

**FOR TIME ANCHOR/PERSONAL INTRODUCTION BULLETS:**
- Focus on project management, scheduling, personal goals
- Ask about deadlines, meeting coordination, project planning
- NO code blocks, NO programming solutions, NO technical implementations
- Use natural conversation about timing, goals, and context

**CODE COMPLEXITY REQUIREMENTS (for all other bullets):**
- **Minimum 20-60+ lines per code block**
- **Multiple functions/methods/classes (4-6 minimum)**
- **Realistic imports and dependencies (3-5 minimum)**
- **Proper error handling, validation, edge cases**
- **Complex business logic, database operations, API calls**
- **Production-level structure and realistic variable names**

**REQUIRED PATTERNS (for coding bullets only):**
- **Debugging (40%):** Generate buggy code with realistic, hard-to-spot errors
- **Code Review (25%):** Generate working but suboptimal code needing improvements
- **Implementation (20%):** Generate partial implementations with detailed TODOs
- **Optimization (15%):** Generate slow/inefficient but functional code

**NEVER use simple examples or tutorial-style code - always production-level complexity**

### 6. AUTHENTIC DEVELOPER STYLE
**Language:** Use contractions ("I'm", "don't"), dev slang ("lol", "btw"), fillers ("like", "um"), informal punctuation ("...", "??", "!!")
**Emotion:** Show genuine feelings - frustration with bugs, excitement about features
**Natural Flow:** Mix question lengths, include rambling, thinking out loud
**Technical Authenticity:** Include actual error messages, file names, version numbers from bullets

### 7. CHRONOLOGICAL ORDER
Process bullet points in exact order provided. Earlier bullet details appear in earlier questions.

### 8. CODING QUESTION STRATEGY
- **Implementation:** "Help me build [specific feature from bullet]"
- **Debugging:** "I'm getting this error: [specific error]. How do I fix it?"
- **Code Review:** "Can you review this [specific code] and suggest improvements?"
- **Optimization:** "How can I make [specific implementation] faster?"

## OUTPUT REQUIREMENTS:
Generate exactly <SUB_BATCH_SIZE> questions that:

1. **MANDATORY:** Use EVERY SINGLE detail from FOCUSED_BULLETS at least once (names, versions, errors, files, specs, etc.)
2. **85% MUST include substantial code snippets (20-60+ lines) with production-level complexity**
3. Sound like genuine developer requests with realistic technical scenarios
4. Follow chronological order of bullet points
5. **ALWAYS include complete, complex code - NEVER use simple examples**
6. Match one of the four coding patterns with appropriate complexity
7. Stay strictly within bullet point scope
8. **MANDATORY VERIFICATION:** Before submitting, confirm every extracted detail appears in the questions

**OUTPUT FORMAT:**
- **CRITICAL: Generate ONLY the developer messages, nothing else**
- Do NOT include question numbers, headers, or organizational text
- **MANDATORY: Separate each complete message with "---MESSAGE_SEPARATOR---"
- Each message can span multiple lines and include code blocks
- **CRITICAL: Each message MUST end with "->-> [number]" where [number] is the bullet point it's based on**
- **MANDATORY: End with "### COMPLETE ###"

**REQUIRED FORMAT PATTERN:**
[Output format example]

**CRITICAL FORMATTING RULES:**
[Output formatting rules]

**CRITICAL VERIFICATION:** Before each question:
1. **MANDATORY FIRST CHECK: What type of bullet is this?
- Time Anchor bullet -> Generate project management/scheduling questions, ABSOLUTELY NO CODE
- Personal Introduction bullet -> Generate career/personal questions FROM their perspective, ABSOLUTELY NO CODE
- Technical bullet -> Generate coding questions WITH substantial code
2. Does this use a specific detail from current bullets?
3. Have I included substantial, complex code (not simple examples) for technical bullets?
4. Does this sound like a real developer asking for help?
5. Am I following one of the four coding patterns correctly for technical bullets?

**FINAL VERIFICATION BEFORE SUBMITTING:**
Count how many extracted details appear in your questions. It MUST be 100% of all details from CURRENT FOCUS AREAS.

```