Secure Coding Checklist

1. Input Validation

Practice	Verified	Notes
Validate all input from untrusted sources (user input, APIs, files)	☐	
Use whitelist validation instead of blacklist	☐	
Validate data type, length, format, and range	☐	
Sanitize input before use in SQL, HTML, XML, OS commands	☐	
Use parameterized queries/prepared statements for SQL	☐	
Validate file uploads (type, size, content)	☐	

2. Output Encoding

Practice	Verified	Notes
Encode output based on context (HTML, JavaScript, URL, CSS)	☐	
Use framework-provided encoding functions	☐	
Implement Content Security Policy (CSP) headers	☐	
Prevent XSS by sanitizing dynamic content	☐	

3. Authentication & Cryptography

Practice	Verified	Notes
Never store passwords in plaintext or using weak hashing	☐	
Use approved cryptographic algorithms (AES-256, RSA-2048+)	☐	
Never hardcode credentials, API keys, or secrets in code	☐	
Use secure random number generators for	☐	

Practice	Verified	Notes
tokens/IVs		
Implement proper certificate validation for HTTPS	☐	

4. Error Handling & Logging

Practice	Verified	Notes
Display generic error messages to users	☐	
Log detailed errors server-side for debugging	☐	
Never expose stack traces or system details to users	☐	
Log security events (auth failures, access violations)	☐	
Never log sensitive data (passwords, tokens, PII)	☐	
Implement centralized error handling	☐	

5. Secure Configuration

Practice	Verified	Notes
Disable debug/verbose modes in production	☐	
Remove default accounts and change default passwords	☐	
Disable directory listing and unnecessary services	☐	
Keep frameworks and dependencies up to date	☐	
Use environment variables for configuration	☐	
Implement security headers (HSTS, X-Frame-Options, CSP)	☐	

6. Dependency Management

Practice	Verified	Notes
Maintain inventory of all dependencies and versions	☐	
Scan dependencies for known vulnerabilities (SCA tools)	☐	
Only use dependencies from trusted sources	☐	
Remove unused dependencies	☐	
Automate dependency updates with testing	☐	