APPS BY VIKAS

Meet Vikas



GenzLudo

Ludo with 3D characters and up to 8 players. A first-of-its-kind no-teaming-up feature keeps play fair. Try it - you'll love the difference.

Download on Google Play



Kachu

Too much scrolling straining your neck? Add your addictive apps and complete daily neck exercises to unlock them.

Download on Google Play

01 / SECRETS & ACCESS

Complete Website & App Security Checklist

27 important security checks explained in easy English

By Vikas Yadav | October 2026

Use this list to review your website or app. Mark a check only after it has been tested. No checklist or AI prompt can guarantee that a website will never be hacked.

- [ ] **1. Hide API Keys and Secrets**

  Check if any secret API key or password is visible in my website code. Hackers could steal these keys and misuse my services.

- [ ] **2. Check Environment Variables**

  Make sure secret values are stored safely and are not exposed to visitors or included in public files.

- [ ] **3. Check Git History for Leaks**

  Find passwords, API keys or private information accidentally uploaded to GitHub in the past. Replace any leaked secrets.

- [ ] **4. Protect Admin Pages**

  Make sure normal users cannot open admin pages or use admin features without permission.

- [ ] **5. Add Secure Authentication**

  Check the login and signup system. Stop attackers from logging in without valid credentials.

- [ ] **6. Check User Permissions**

  Make sure users can only see or change their own data. Stop one user from accessing another user's private information.

02 / SESSIONS, INPUTS & APIS

Sessions, Inputs & Apis

- [ ] **7. Secure Login Sessions and Tokens**

  Protect login cookies and tokens from being stolen or misused. Make sure logout and session expiry work properly.

- [ ] **8. Validate All Form Inputs**

  Check every form and API input. Reject invalid or dangerous data before it reaches the server.

- [ ] **9. Protect Against XSS Attacks**

  Stop attackers from injecting harmful JavaScript through forms, comments or other user content.

- [ ] **10. Prevent SQL Injection**

  Make sure attackers cannot use form inputs or API requests to run harmful database queries.

- [ ] **11. Add CSRF Protection**

  Stop malicious websites from tricking logged-in users into performing unwanted actions. Check protection where cookies are used for login.

- [ ] **12. Add Rate Limiting**

  Stop people or bots from sending too many requests. This helps prevent login attacks, spam and API abuse.

- [ ] **13. Secure All API Endpoints**

  Check every API route for missing login checks, weak permissions and unsafe inputs. Stop unauthorized access.

03 / BROWSER & PRODUCTION SAFETY

Browser & Production Safety

- [ ] **14. Check CORS Settings**

  Make sure browser access from other websites follows the correct rules. Avoid unsafe settings that could expose private responses.

- [ ] **15. Add Security Headers**

  Add headers that help protect the website against harmful scripts, unsafe browser behavior and certain attacks.

- [ ] **16. Enforce HTTPS**

  Make sure data travels through a secure connection. This helps protect passwords and private information during transfer.

- [ ] **17. Disable Debug Mode**

  Turn off development features and detailed error messages in production. Stop exposing technical information that attackers could misuse.

- [ ] **18. Update Dependencies**

  Find old or vulnerable libraries and packages. Update them safely without breaking existing features.

- [ ] **19. Remove Unnecessary Code**

  Find and remove unused packages, test routes, old features and unnecessary services that could create security risks.

- [ ] **20. Check for Exposed Files**

  Make sure .env files, database backups, private documents and sensitive configuration files are not publicly accessible.

04 / DATA, PAYMENTS & RECOVERY

Data, Payments & Recovery

- [ ] **21. Secure the Database**

  Check database access, permissions and connections. Stop unauthorized people from reading, changing or deleting private data.

- [ ] **22. Hash Passwords Properly**

  Make sure passwords are never stored as plain text. Use strong password hashing like Argon2id.

- [ ] **23. Secure File Uploads**

  Check uploaded files for dangerous types, fake extensions and oversized content. Stop attackers from uploading harmful files.

- [ ] **24. Secure Payments and Webhooks**

  Verify payment information on the server and check webhook signatures. Stop fake payment confirmations and unauthorized actions.

- [ ] **25. Prevent Unsafe Server Requests**

  Stop attackers from making the server access private systems or unsafe addresses through user-provided URLs.

- [ ] **26. Add Security Monitoring and Backups**

  Detect suspicious logins, errors and unusual activity. Keep secure backups so important data can be recovered after an incident.

- [ ] **27. Run a Full Security Audit**

  Scan the entire application for security problems. Fix the issues, test the changes and check again to confirm the fixes work.

05 / COPY & USE

Final Instruction for Claude

Paste this after the 27 checks when asking Claude to review your app.

Do not just tell me my website is secure.

Check every security point carefully and show me what problems you found.

Fix the problems without breaking my website or changing how it works.

Test every fix and run the security checks again.

If something cannot be verified then clearly tell me.

Do not guess and do not claim my website is 100% secure.

For your video

You can use these 27 points as a complete security checklist. They cover many common ways websites and apps get attacked, but no checklist or AI prompt can guarantee that a website will never be hacked.

More things worth exploring

Find more free guides at vikasbuilds.com/resources