1. Add account points/currency to DB
   - Verify DB table + API read/write path
   - New column/field exists for points. New user should have 0 points.
   - Points update globally after refresh
   - Screenshot of DB schema, user rows before & after points update

2. Add points animation
   - When points are awarded, animation triggers exactly once per award event
   - Verify the Event + UI when points are awarded
   - 5 second Video-clip of successful award → animation

3. Add points counter to user page
   - user page needs to show points (verify User/Profile page)
   - refreshing page shows persistent values
   - Screenshot of profile page with correct points visible/ points changing (before & after)

4. Add points counter to toolbar
   - Verify point counter on toolbar, and verify it on global nav bar/tool bar
   - Points are visible on every page where toolbar appears
   - No overflow display on large values (999+ points)
   - Screenshot of toolbar showing functioning points

5. Add admin tool to add points
   - Verify admin page + permissions
   - Admin can select a user + enter amount + submit
   - User's points update in DB and reflect in UI
   - Screenshot/video of admin adding points and user points increasing
   - Screenshot of non-admin access blocked

6. Award points for completing quests
   - Verifying the question completion lifecycle
   - Completing a quest triggers points award
   - Same quest cannot be exploited repeatedly
   - Screen recording: complete quest → points increase → refresh still increased
7. Add profile pictures functionality
   - Screenshot: profile picture displayed
   - users can set/change profile pictures
   - Selected picture persists after refresh/re-login

8. Create initial set of profile pictures
   - verify initial set in folder
   - At least a number of profile picture are set
   - Screenshot of the folder directory containing profile pictures (names visible)

9. Add tab for profile picture shop
   - verify on nav bar + shop page
   - Shop displays available pictures + price
   - Screenshot/video of a “purchase/select” action and resulting change
   - Screenshot of shop page functioning successfully
   -

10. Create Tab for Campus Guesser game
    - Nav tab + route to Campus Guesser landing screen
    - screenshot of tab + landing screen.
    -

11. Create database of KU campus photos with correct location metadata
    - DB table schema + sample rows (photo id/url + lat/long + tags + difficulty)

12. Build game interface displaying random campus photo
    - Interface that displayed random photos of campus via JPEG, PNG, etc.
    - Gameplay screen showing random photo render; short video demonstration

13. Create interactive KU campus map for user guessing
    - interactive and dynamic map, 1:1 to KU campus
    - Map UI with clickable guess placement; screenshot showing pin drop + coordinates captured
14. Implement scoring system based on guess proximity
    - Scoring formula documented + example calculation (two sample guesses) + UI showing score result
15. Add round timer with configurable time limits
    - Timer visible + settings control + proof it ends round at 0 via short screen recording.

16. Display correct location after each guess with accurate distance info
    - Post-guess results screen: correct pin + user pin + distance label.

17. Create multi-round game structure (5 rounds per game
    - Round counter + transitions; video showing rounds 1→5.

18. Calculate and display total score at end of game
    - End screen with total score + per round breakdown

19. Link CampusGuesser scores to existing user account points
    - After game end, points increase in user wallet; screenshot DB + UI.

20. Add CampusGuesser achievements to user profile
    - Achievement earned event + profile display.
    -

21. Update Side Quest homepage to feature CampusGuesser
    - Add CampusGuesser section/card on homepage
    - Add CTA button (“ Play CampusGuesser”)
    - CTA routes to CampusGuesser landing page
22. Create admin interface for adding new campus photos
    - Admin-only upload form (image and metadata)
    - Submitting creates DB row successfully
    - Non-admin user are blocked from access
23. Implement photo tagging system (buildings, landmarks, etc.)
    - Add tags field to photos in DB
    - Admin can add/edit tags per photo
    - Admin can filter/search photos by tag
24. Add difficulty rating to photos (easy/medium/hard)
    - Add difficulty rating to photos (easy/medium/hard)
    - Admin can set difficulty on upload/edit
    - Game can pull photos by selected difficulty
25. Create photo verification system to ensure accuracy
    - Add status: pending/approved/rejected
    - New uploads default to pending
    - Game only uses approved photos
26. Implement difficulty levels (different campus areas / photo pools)
    - Add difficulty selection before starting
    - Each difficulty uses a different photo pool
    - Different selections produce different photo sets
27. Add hint system (reveal building/general area)
    - Add hint button on gameplay screen
    - Hint reveals building/general area info
    - Add rule: cooldown or point cost
28. Create timed challenge mode with speed bonuses
    - Add mode selector (Normal vs Timed)
    - Timed mode adds countdown + ends at 0
    - Bonus points scale with time remaining
29. Add daily challenge with specific photo set
    - Add “Today’s Challenge” entry point
    - Daily photo set is deterministic for all users
    - Daily set changes based on date (server-defined)
30. Create CampusGuesser leaderboard (daily/weekly/all-time)
    - Leaderboard page with tabs (daily/weekly/all-time)
    - Sort by score (desc) with tie-break rule
    - Shows usernames + scores + ranks
31. Add animations for leaderboard records
    - Animate rows on load/new entry
    - Animate rank changes (up/down)
    - Animation triggers once per update

32. Implement share score to social media feature
    - Add “Share” button on results screen
    - Generate share text + link (or image card)
    - Open share modal / copy-to-clipboard
33. Update map detail for CampusGuesser
    - Improve map styling (labels/paths/landmarks)
    - Constrain bounds to KU campus area
    - Keep map performance smooth
34. Design responsive game interface for mobile devices
    - Layout adapts to mobile widths cleanly
    - No overflow for timer/buttons/map
    - Works on at least 2 common viewport sizes
35. Add tutorial/onboarding for first-time players
    - First-run tutorial overlay walkthrough
    - “Don’t show again” toggle
    - Persist tutorial completion state
36. Implement visual feedback for guess accuracy (heatmap/ring)
    - Show accuracy visual after each guess
    - Include legend for meaning of colors/rings
    - Keep distance label visible too
37. Create game statistics page (accuracy, favorite locations)
    - Add stats page linked from profile/game
    - Track 3+ metrics (avg distance, best score, games played)
    - Show “favorite locations/tags” summary
38. Set up Supabase tables for CampusGuesser game data
    - Create tables for games/rounds/guesses/photos
    - Add FKs + indexes for key relationships
    - Store enough data to rebuild results history
39. Create API endpoints for game logic and scoring
    - Endpoints: start game, get photo, submit guess, end game
    - Validate auth + inputs server-side
    - Return consistent JSON + status codes
40. Implement photo randomization algorithm avoiding repeats
    - Use shuffle/bag method per game session
    - Guarantee no repeats within a 5-round game
    - Reset only on new game or bag exhaustion