

1. App structure & navigation

Primary navigation (BottomNavigationBar, persistent)

- Home (Dashboard)
- Attendance
- Tasks
- Notifications
- Profile

Notes: Use Flutter's `BottomNavigationBar` or a custom `ConvexAppBar` for a modern look. Keep navigation simple and predictable.

2. Authentication

Screen: Login

- Fields: Mobile number (10 digit), Password
- CTA: Sign in
- Footer links: Forgot password, Help

UX notes for frontend

- Validate inputs client-side (formatting for phone number, password min length).
- Store auth tokens securely on the device using `flutter_secure_storage` (no backend instructions here — just storage method).

3. Dashboard (Home)

Purpose: quick overview and shortcuts

Top area: Greeting, date, current city (derived from device location)

Cards / Tiles:

- Today's attendance status (Not punched / Punched at 09:02)
- Ongoing tasks (count)
- Quick action buttons: Punch In / Punch Out, Create Task
- Recent submissions awaiting review

Design tip: Make Punch In/Out a prominent FloatingActionButton (FAB) for quick access.

4. Attendance Module — UX flows (frontend-focused)

4.1 Punch In / Punch Out

Entry points: Dashboard FAB or Attendance tab

Step 1 — Start Punch

- Button: Mark Attendance
- Microcopy: "You will take a selfie and share your current location"

Step 2 — Selfie capture

- Use `camera` package in Flutter for camera preview and capture.
- UI: rounded live preview, camera toggle (front/back), flash.
- Options: Capture, Retake, Use Photo from Gallery (use `image_picker`).

Step 3 — Location & Confirmation

- After capture, show a confirmation screen with:
 - Selfie thumbnail (tap to enlarge)
 - Detected address (reverse geocoded locally or via an SDK)
 - Small map preview (use `google_maps_flutter` or `flutter_map`)
 - Timestamp (derived on device)
 - Button: Confirm & Punch In

On success

- Show a success animation (Lottie or simple checkmark) and haptic feedback.

Data collected (frontend state): userId, timestamp, selfieFilePath, lat/lon, accuracy, deviceId, punchType (IN/OUT)

4.2 Attendance History (UI)

Two view modes: Calendar view and List view

Calendar view:

- Month grid using a calendar package (e.g., `table_calendar`)
- Day cells colored for status; tap opens day detail modal with punch data and selfie preview

List view:

- Vertical `ListView` of attendance cards with date, IN/OUT times, total hours, and small selfie avatar; include filters and month picker.

5. Task Tracking Module — UX flows (frontend-focused)

5.1 Projects list (entry)

Screen: Projects displayed as cards (title, brief, client, project status chip)

- Card contents (reference from website): Project title, client name (subtext), short description, start / end dates, status badge (Planned / In Progress / Completed).
- UI: `ListView` of `Card` widgets; each card shows an action to open Project Details.
- Controls: Search bar, Filter chips (All / My Projects / In Progress / Planned / Completed), Sort by Recent.

5.2 Project detail

Header: Project title, client, status badge.

Sections (tabs): Tasks | Resources | Files

CTA: Create Task (sticky floating button)

Edit Project: (admin-only can see this option)

- Fields: Client (dropdown), Project Name, Description, Start Date, End Date, Status.

5.3 Task list

Table-like mobile adaptation:

- Each task shown as a compact card with:
 - Task Creation Date (e.g., Nov 01, 2025)
 - Project title
 - Short description (one-line)
 - Time taken (e.g., 480 min)
 - Status chip (Completed / Approved / In Progress / Pending Approval)
 - Resource cost (if available)
 - Action icon to view full Task Detail

Search & Filters: search by project and status filters (All / Pending / Approved / Completed / Rejected).

5.4 Create Task (form)

Open as full-screen route or modal. Fields (based on website):

- **Task Name** (required)
- **Description** (multiline)
- **Date** (DatePicker; default: today)
- **Location** (auto-detected with option to edit; show reverse-geocoded address and small map preview)

- **Time Taken** (duration input in minutes with an optional human-readable display like `480 minutes (8.0 hrs)`)
- **Project** (select from projects; preselected if opened from within a project)

Attachments (Photos & Files)

- Horizontal thumbnail stack showing attachments (images and documents) with the following actions on each thumbnail:
 - Preview (tap to open full-screen viewer)
 - Delete
 - Reorder (drag handle)
- Add button opens bottom sheet: Camera / Gallery / Files (use `image_picker` for images and `file_picker` for documents).
- Show file size below filename on the attachment chip (e.g., 2.3 MB).
- For images: show a small camera icon overlay; for PDFs show a document icon.

Resources Used (table-like inline list — mirrors web Task detail table)

- Employees should be able to enter resources used while doing a particular task
- Each row fields:
 - Resource Name (autocomplete or text)
 - Quantity (number input)
 - Unit (text, e.g., pcs, m)
 - Unit Cost (numeric input) — optional for field entry; if filled, compute total cost
 - Total Cost (computed)
 - Remove action (trash)
- Footer: Total Resource Cost (sum of row totals)
- UX: by default a single empty row shown; user can add rows using `+ Add Resource` button.

Buttons / Actions

- Submit Task

Validation

- Require: Task Name, Date. For fields like Time Taken, show help text telling users to enter minutes or pick duration.
- If no network: show `Save Draft` and `Submit (Queued)` states with clear sync indicators.

5.5 Task Detail

Header: Task title and status badge (e.g., Completed / Pending Review)

Employee should be able to change status of the task (Draft, ToDo, InProgress, Completed)

Task Information card: display two-column style stacked vertically on mobile with:

- Date
- Time Taken (show both minutes and human-readable hours)
- Location (address + tap to open map)
- Project
- Description

Attachments section:

- Thumbnails / file chips showing name and size (e.g., cable_routing_plan.pdf — 1000.0 KB)
- Tap opens file viewer or image preview with swipe left/right to move between attachments.
- Employee should be able to attach attachments (photos, pdf)

Resources Used section:

- Vertical list showing for each resource: name, quantity + unit, unit cost (if available), total cost, and a trash icon for admins/editors.
- Employee should be able to edit the resources, or add resources.
- At the bottom: Total Resource Cost — bold and prominent.

Important Points to Note:

- Make Whole UI really really attractive and professional
- Do not waste credits in creating .md files
- **Micro-interactions:**
 - Camera: shutter animation and short haptic pulse
 - Punch success: animated checkmark + short vibration
 - Image uploads & sync: progress indicator on thumbnails
 - Field errors: inline messages and gentle shake animation
 - Empty states: friendly illustrations and CTAs
- **Attachments — UX & Implementation details (mobile)**
 - Use `image_picker` for camera/gallery; `file_picker` for PDFs and other documents.

- Compress images on-device (`flutter_image_compress`) before storing locally to save space.
- Store attachments temporarily in app storage and keep a thumbnail cache for list screens.
- Show file size and upload status on each thumbnail (Draft / Queued / Uploading / Synced / Failed).
- Provide previewer using `photo_view` for images and a PDF viewer for documents