Below is a **detailed task implementation guide** for the **UI/UX Designer** working on the **VerseProjection SaaS web app** MVP, focused on real-time Bible verse auto-detection and projection using public-domain KJV/WEB texts. This guide aligns with the PRD, emphasizing the continuous speech analysis, top 10 verse matching, user selection, and projection workflow. The guide provides comprehensive instructions for designing the **Login Page**, **Admin Dashboard**, **Projection Window**, and **Error Notifications**, as well as the **Electron-based offline app UI**, ensuring an intuitive, accessible, and efficient experience for non-technical church volunteers (primary persona: ages 25–50, moderate technical proficiency). The design must support <2-second end-to-end latency, WCAG 2.1 accessibility, and compatibility with Chrome/Firefox browsers and 720p/1080p projectors, all within the 6–9-month timeline and $150K–$400K budget.

Task Implementation Guide for UI/UX Designer: VerseProjection MVP

1. Overview
Role: UI/UX Designer
Objective: Design a user-friendly, responsive, and accessible interface for the VerseProjection MVP, enabling church tech volunteers to configure settings, select from top 10 verse matches, override misdetections, and project verses seamlessly. The UI must support real-time speech analysis, user feedback for model learning, and offline functionality via an Electron app.
Scope:
- **Web App**: Login Page, Admin Dashboard (settings, transcription view, matches table, override search, detection history), Projection Window, Error Notifications.
- **Offline App**: Electron-based UI mirroring web app functionality.
- **Deliverables**: Wireframes, high-fidelity mockups, interactive prototypes (Figma), design system, accessibility audit, user testing plan.
Timeline: Months 5–8 (development phase, per PRD), with deliverables integrated into frontend development (React, Material-UI/Tailwind CSS).
Budget Allocation: ~$20K–$40K (part of frontend development, included in $150K–$400K).

Tools: Figma (design/prototyping), Adobe XD (optional), WCAG 2.1 checker (e.g., WAVE), GitHub (asset sharing with developers).

2. Design Principles
- **Simplicity**: Minimize clicks (e.g., one-click projection start, single-click verse selection) for non-technical users.
- **Reliability**: Provide clear, real-time feedback (transcription, matches, errors) to build trust.
- **Accessibility**: Comply with WCAG 2.1 Level AA (e.g., high-contrast text, keyboard navigation, screen reader support).
- **Responsiveness**: Support laptop screens (1366x768 to 1920x1080) and projectors (720p, 1080p).
- **Consistency**: Use a cohesive design system (colors, typography, components) across web and Electron apps.
- **Performance**: Ensure UI elements load/render in <200ms to maintain <2-second latency.

3. Design System
Create a **design system** in Figma to standardize UI components, ensuring consistency and developer handoff efficiency.
- **Typography**:
  - Primary: Roboto (open-source, readable, supports WCAG contrast).
        - Headings: Roboto Bold, 24–36pt.
        - Body: Roboto Regular, 14–18pt.
        - Verse Text: Roboto Medium, 12–48pt (user-configurable).
  - Fallback: Arial (widely available).
- **Colors**:
  - Primary: #1976D2 (blue, calming, church-appropriate).
  - Secondary: #388E3C (green, for success/selection).
  - Background: #F5F5F5 (light gray, neutral) or #121212 (dark mode).
  - Text: #212121 (dark, 4.5:1 contrast on light) or #FFFFFF (white, on dark).
  - Error: #D32F2F (red, alerts).
  - WCAG Compliance: Ensure 4.5:1 contrast for text, 3:1 for UI elements.
- **Icons**: Material-UI icons (e.g., `Mic`, `Search`, `PlayArrow`, `Settings`).
- **Spacing**: 8px grid system (e.g., 8px, 16px, 24px margins/padding).
- **Components**:
  - Buttons: Primary (filled, blue), Secondary (outlined, green), Disabled (gray).

  - Inputs: Text fields, dropdowns, sliders (Material-UI styles).
  - Tables: Sortable, clickable rows for match selection.
  - Modals: For errors, confirmation dialogs.
- **Dark/Light Modes**: Toggleable themes (default: light for churches).
- **Export**: Provide Figma tokens (colors, typography, spacing) for Tailwind CSS integration.

4. UI Components and Tasks
Design the following components, ensuring each supports the PRD's functional requirements (continuous analysis, top 10 matches, user selection, projection, learning).

4.1 Login Page
Purpose: Authenticate users (email/password or SSO via AWS Cognito) with a clean, minimal interface.
Requirements:
- Support email/password login and SSO (e.g., Google, Microsoft).
- Responsive for 1366x768 to 1920x1080 laptop screens.
- <200ms load time, WCAG 2.1 compliant.
- Include dark/light mode toggle.

Design Tasks:
1. **Wireframe**:
   - Layout: Centered card (400x500px) with logo, email field, password field, SSO buttons, "Login" button, dark/light toggle (top-right).
   - Footer: "Forgot Password?" link, app version (e.g., "v1.0.0").
2. **Mockup**:
   - Card: White (#FFFFFF) or dark (#121212), 16px border-radius, subtle shadow (0 2px 4px rgba(0,0,0,0.1)).
   - Logo: VerseProjection placeholder (100x50px, top of card).
   - Email/Password Fields: Material-UI TextField (full-width, 14pt Roboto, placeholder: "email@example.com").
   - SSO Buttons: Outlined buttons ("Sign in with Google", "Sign in with Microsoft"), 48px height, 16px icons.
   - Login Button: Primary blue (#1976D2), filled, 48px height, "Login" (Roboto Bold, 16pt).
   - Dark/Light Toggle: Material-UI Switch (top-right, 32px).

   - Error State: Red text below fields (e.g., “Invalid credentials”), 14pt, #D32F2F.
3. **Interactions**:
   - On submit: Show loading spinner (Material-UI CircularProgress, centered).
   - On error: Display error text, shake card (200ms animation).
   - Keyboard: Support Enter key for login, Tab for field navigation.
   - Accessibility: ARIA labels (e.g., `aria-label="Email input"`), 4.5:1 contrast.
4. **Prototype**:
   - Simulate login flow: Enter credentials → Spinner → Dashboard (success) or Error (failure).
   - Test SSO button clicks (redirect placeholder).
5. **Handoff**: Export assets (logo, icons), provide Figma link with component specs (padding: 16px, font sizes, colors).

Deliverables: Wireframe, mockup, prototype (Figma), accessibility checklist (WCAG 2.1).

4.2 Admin Dashboard
Purpose: Central hub for configuring settings, viewing real-time transcription, selecting from top 10 verse matches, overriding misdetections, and monitoring detection history.
Requirements:
- Real-time updates via WebSocket (transcription, matches).
- Clickable table for top 10 verse matches (95%+ correct verse inclusion).
- Manual override search bar with autocomplete.
- Intuitive for non-technical users, <200ms UI updates.
- Responsive (1366x768 to 1920x1080), WCAG 2.1 compliant.

Design Tasks:
1. **Wireframe**:
   - Layout: Sidebar (200px, settings), Main Panel (flexible, split into 4 sections).
   - Sidebar:
      - Menu: Settings, Audio Input, Display, Confidence Threshold, Logout.
      - Start/Stop Projection button (prominent).
   - Main Panel:
      - Transcription View (top, 20% height): Real-time speech text.
      - Matches Table (middle, 50%): Top 10 verses (Reference, Text, Version, Confidence).

- Override Search Bar (below table, 10%): Autocomplete input.
- Detection History (bottom, 20%): Timeline of matches/selections.

2. **Mockup**:
   - **Sidebar**:
     - Background: #E0E0E0 (light) or #1E1E1E (dark).
     - Menu Items: 16pt Roboto, icons (e.g., `Settings`, `Mic`), hover effect (background: #1976D2, white text).
     - Start/Stop Button:
     - Start: Green (#388E3C), filled, “Start Projection” (Roboto Bold, 16pt), `PlayArrow` icon.
     - Stop: Red (#D32F2F), outlined, “Stop Projection”, `Stop` icon.
     - Size: 48px height, 100% width.
   - **Settings Panel** (collapsible, triggered by Sidebar):
     - Bible Version: Dropdown (KJV, WEB), 14pt Roboto.
     - Audio Input: Dropdown (e.g., “USB Mic”, “Soundboard Line-In”), test button (`Mic` icon, triggers 5s audio preview).
     - Display:
     - Font Size Slider: 12–48pt, step 2pt.
     - Color Picker: Text (default: #FFFFFF), Background (default: #000000).
     - Font Family: Dropdown (Roboto, Arial, Times New Roman).
     - Confidence Threshold: Slider (0.7–0.9, step 0.05, default: 0.7).
     - Save Button: Blue (#1976D2), “Save Settings”, 48px height.
   - **Transcription View**:
     - Background: #FFFFFF or #121212, 16px padding.
     - Text: 14pt Roboto, scrollable (10s history, max 5 lines).
     - Example: “Pastor: ‘God so loved the world’” (updated every 2s).
     - Border: 1px solid #E0E0E0, 8px radius.
   - **Matches Table**:
     - Columns: Reference (15%, e.g., “John 3:16”), Text (50%, truncated to 100 chars), Version (15%, e.g., “KJV”), Confidence (20%, e.g., “92%”).
     - Rows: 10 max, 48px height, hover effect (background: #BBDEFB).
     - Selection: Click row → Green border (#388E3C), projects verse.
     - Sorting: Default by Confidence (descending), clickable headers.
     - Style: Material-UI DataGrid, 14pt Roboto, alternating row colors (#FFFFFF, #F5F5F5).
   - **Override Search Bar**:
     - Material-UI TextField, 100% width, 14pt Roboto.

        - Placeholder: “Search verse (e.g., John 3:16)”.
        - Autocomplete: Dropdown with verse suggestions (e.g., type “Jo” → “John 1:1”, “John 3:16”).
        - Search Button: Blue (#1976D2), `Search` icon, 48px height.
  - **Detection History**:
        - Timeline format: Timestamp (e.g., “10:02 AM”), Event (e.g., “John 3:16 selected”), Confidence.
        - Scrollable, max 20 entries, 14pt Roboto.
        - Clickable: Re-project past selection.
        - Border: 1px solid #E0E0E0, 8px radius.
3. **Interactions**:
  - **Start/Stop**: Click “Start Projection” → Button toggles to “Stop”, transcription/matches activate.
  - **Matches**: Click row → Verse projects, row highlights green, selection logged.
  - **Override**: Type in search bar → Autocomplete suggests verses → Enter or click “Search” → Verse projects.
  - **Settings**: Adjust sliders/dropdowns → Real-time preview in Projection Window (e.g., font size).
  - **Keyboard**:
        - Arrow keys: Navigate Matches Table rows.
        - Enter: Select highlighted row or search result.
        - Tab: Cycle through inputs/buttons.
  - **Accessibility**:
        - ARIA labels (e.g., `aria-label="Top verse matches table"`).
        - Screen reader support for transcription and matches (e.g., “Match 1: John 3:16, 92% confidence”).
        - 4.5:1 contrast for text, 3:1 for buttons/icons.
4. **Prototype**:
  - Simulate flow: Start Projection → Transcription updates (e.g., “God so loved”) → Matches Table populates (10 verses) → Click row → Verse projects → Override search → New verse projects.
  - Test settings changes (e.g., font size, threshold) and error states (e.g., no matches).
5. **Handoff**:
  - Export assets: Icons, table styles, button states.

- Provide Figma link with component specs (e.g., table row: 48px height, 16px padding).
- Document interactions (e.g., hover: #BBDEFB, click: green border).

Deliverables: Wireframe, mockup, prototype (Figma), accessibility checklist, design system tokens.

4.3 Projection Window
Purpose: Display user-selected verse in a full-screen browser window for projector output (HDMI/VGA).
Requirements:
- Customizable display (font size: 12–48pt, color, background, font family).
- Smooth transitions (<200ms fade-in).
- Compatible with 720p (1280x720) and 1080p (1920x1080) projectors.
- WCAG 2.1 compliant (high-contrast text).

Design Tasks:
1. **Wireframe**:
   - Layout: Full-screen, centered verse text with reference (e.g., “John 3:16”).
   - Optional: Version label (e.g., “KJV”) at bottom.
2. **Mockup**:
   - Background: User-configurable (default: #000000, black).
   - Text:
        - Verse: Roboto Medium, 12–48pt (default: 24pt), user-configurable color (default: #FFFFFF).
        - Reference: Roboto Bold, 80% of verse size (e.g., 19pt), same color.
        - Version: Roboto Regular, 50% of verse size (e.g., 12pt), bottom-right.
        - Alignment: Centered, 16px padding from edges.
   - Transition: 200ms fade-in for new verses.
   - Contrast: Ensure 4.5:1 (e.g., white on black: 21:1).
3. **Interactions**:
   - On selection: Verse fades in (200ms), replacing previous verse (if any).
   - On settings change: Update font size/color in real-time (e.g., slider moved in dashboard).
   - Accessibility:
        - ARIA live region (`aria-live="polite"`) for screen readers (e.g., “Projected: John 3:16”).

- High-contrast mode enforced (user cannot select low-contrast combinations).
4. **Prototype**:
   - Simulate verse transitions: Select “John 3:16” → Fade-in → Select “Romans 8:28” → Fade-in.
   - Test settings: Adjust font size (12pt to 48pt), color, background.
   - Verify resolutions: 720p (1280x720), 1080p (1920x1080).
5. **Handoff**:
   - Export text styles (font sizes, weights, colors).
   - Provide Figma link with specs (e.g., text alignment: center, padding: 16px).
   - Document transition (200ms fade, CSS: `opacity: 0 to 1`).

Deliverables: Wireframe, mockup, prototype (Figma), accessibility checklist.

4.4 Error Notifications
Purpose: Alert users to issues (e.g., low audio quality, no matches) with clear actions (e.g., check mic, use override).
Requirements:
- Non-intrusive, dismissible pop-ups.
- Clear messaging for non-technical users.
- WCAG 2.1 compliant, <200ms display time.

Design Tasks:
1. **Wireframe**:
   - Layout: Modal (300x200px, centered) with title, message, action button(s), close button.
   - Position: Overlays dashboard, dimmed background (50% opacity).
2. **Mockup**:
   - Modal:
      - Background: #FFFFFF or #121212, 16px border-radius, shadow (0 4px 8px rgba(0,0,0,0.2)).
      - Title: 16pt Roboto Bold, #D32F2F (e.g., “Audio Issue”).
      - Message: 14pt Roboto, #212121 or #FFFFFF (e.g., “Low audio quality. Check microphone or select input.”).
      - Buttons:
      - Primary: Blue (#1976D2), “Check Audio” or “Use Override”, 48px height.
      - Close: Gray (#757575), “Close”, 48px height.

    - Timeout: Auto-dismiss after 10s (configurable).
  - Examples:
    - “No matches found. Confidence below 70%. Try manual override.” (Button: “Open Search”).
    - “Transcription failed. Reconnect microphone.” (Button: “Check Audio”).
3. **Interactions**:
  - On error: Modal appears, background dims, focus on primary button.
  - On click:
    - “Check Audio” → Opens audio input dropdown.
    - “Open Search” → Focuses override search bar.
    - “Close” → Dismisses modal.
  - Keyboard:
    - Esc: Close modal.
    - Enter: Trigger primary button.
    - Tab: Navigate buttons.
  - Accessibility:
    - ARIA role (`role="alertdialog"`, `aria-label="Error: Low audio quality"`).
    - Focus management (trap focus in modal).
    - 4.5:1 contrast for text.
4. **Prototype**:
  - Simulate errors: Low audio → Modal → Click “Check Audio” → Dropdown opens.
  - Test dismissal: Click “Close” or wait 10s.
  - Verify focus trapping for accessibility.
5. **Handoff**:
  - Export modal assets (buttons, icons).
  - Provide Figma link with specs (e.g., modal: 300x200px, padding: 16px).
  - Document animations (e.g., 200ms scale-in, CSS: `transform: scale(0.9 to 1)`).

Deliverables: Wireframe, mockup, prototype (Figma), accessibility checklist.

4.5 Offline App (Electron)
Purpose: Mirror web app UI in an Electron-based desktop app for offline use in low-internet environments.
Requirements:

- Identical functionality: Settings, transcription, top 10 matches, override, projection.
- ~600MB disk space, runs on 8GB RAM, Core i5 laptops.
- Sync selections to cloud when online.
- WCAG 2.1 compliant, <200ms UI updates.

Design Tasks:
1. **Wireframe**:
   - Layout: Identical to web app (Sidebar, Main Panel: Transcription, Matches, Override, History).
   - Additional: Sync status indicator (top-right, e.g., “Offline”, “Syncing”).
2. **Mockup**:
   - Reuse web app design system (Roboto, #1976D2, Material-UI components).
   - Sync Indicator:
      - Offline: Gray (#757575), “Offline” with `CloudOff` icon.
      - Syncing: Blue (#1976D2), “Syncing” with spinning `Sync` icon.
      - Synced: Green (#388E3C), “Synced” with `CheckCircle` icon.
      - Position: Top-right, 14pt Roboto, 32px height.
   - Platform-Specific:
      - Windows/macOS title bar (minimize, maximize, close buttons).
      - Menu bar: File (Quit), Edit (Settings), Help (Documentation).
   - Storage Warning: Modal if disk space <600MB (e.g., “Insufficient storage. Free up 600MB.”).
3. **Interactions**:
   - Match web app: Start/Stop, match selection, override, settings.
   - Sync:
      - On reconnect: Auto-sync feedback data (SQLite → PostgreSQL).
      - Show “Syncing” → “Synced” transition (200ms fade).
   - Offline Warning: Modal on launch if no internet (e.g., “Offline mode active. Selections will sync later.”).
   - Keyboard: Identical to web app (arrows, Enter, Tab).
   - Accessibility: Same ARIA labels, focus management, and contrast ratios.
4. **Prototype**:
   - Simulate offline flow: Launch app → Offline warning → Start Projection → Select match → Sync on reconnect.
   - Test platform-specific elements (title bar, menu bar).

  - Verify storage warning modal.
5. **Handoff**:
  - Export assets: Sync icons, platform-specific elements.
  - Provide Figma link with specs (e.g., sync indicator: 32px height, 8px padding).
  - Document differences (e.g., Electron title bar vs. browser).

Deliverables: Wireframe, mockup, prototype (Figma), accessibility checklist.

5. Accessibility Implementation
Ensure WCAG 2.1 Level AA compliance across all components:
- **Contrast**:
  - Text: 4.5:1 (e.g., #212121 on #FFFFFF).
  - UI Elements: 3:1 (e.g., #1976D2 buttons on #F5F5F5).
  - Tool: Use WAVE or Contrast Checker in Figma.
- **Keyboard Navigation**:
  - Tab order: Logical flow (e.g., email → password → login button).
  - Arrow keys: Navigate Matches Table, autocomplete suggestions.
  - Enter: Submit forms, select matches.
  - Esc: Close modals.
  - Test: Manual keyboard testing, no mouse.
- **Screen Readers**:
  - ARIA Labels:
    - Login: `aria-label="Email input"`, `aria-label="Login button"`.
    - Dashboard: `aria-label="Top verse matches table"`, `aria-live="polite"` for transcription.
    - Projection: `aria-live="polite"` for verse updates.
    - Modals: `role="alertdialog"`, `aria-label="Error: Low audio quality"`.
  - Test: Use NVDA (Windows) or VoiceOver (macOS) to verify readability.
- **Focus Management**:
  - Trap focus in modals (first button to last).
  - Highlight focus (e.g., 2px blue outline, #1976D2).
  - Test: Ensure focus remains visible and logical.
- **Text Resizing**: Support browser zoom (up to 200%) without breaking layout.
- **Audit**: Conduct WCAG audit before handoff, document compliance (e.g., “4.5:1 contrast verified for all text”).

Deliverables: Accessibility checklist, audit report, screen reader test results.

6. User Testing Plan
Validate the UI/UX with church volunteers to ensure usability and adoption.
- **Participants**: 10–15 church tech volunteers (ages 25–50, 50–500 member churches, urban/suburban).
- **Tasks**:
  - Log in (email/password or SSO).
  - Configure settings (Bible version, font size, audio input).
  - Start projection, select a verse from top 10 matches.
  - Override a misdetection using search bar.
  - Handle an error (e.g., low audio quality).
  - Use offline app (Electron).
- **Metrics**:
  - Success Rate: 90%+ complete tasks without assistance.
  - Time: <30s to start projection, <5s to select a match.
  - Satisfaction: 80%+ rate UI as “easy to use” (5-point scale).
- **Method**:
  - Remote testing (Zoom, screen sharing) or in-person at pilot churches (Month 8).
  - Record sessions (with consent) for analysis.
  - Use Figma prototype for early testing, React app for final testing.
- **Feedback**:
  - Identify pain points (e.g., confusing match table, slow settings).
  - Prioritize fixes (e.g., larger buttons, clearer labels).
  - Document in GitHub issues for frontend team.
- **Iterations**: 2 rounds (Month 7: prototype, Month 8: React app).

Deliverables: User testing plan, participant script, feedback report, iteration recommendations.

7. Implementation Timeline
- **Month 5**:
  - Create design system (typography, colors, components).
  - Design Login Page wireframe, mockup, prototype.
  - Conduct initial accessibility audit (WCAG 2.1).

- **Month 6**:
  - Design Admin Dashboard (settings, transcription, matches, override, history).
  - Design Projection Window and Error Notifications.
  - Build interactive prototype (Figma, all components).
  - Test prototype with 5 internal users (developers, PM).
- **Month 7**:
  - Design offline Electron app UI, reusing web assets.
  - Conduct user testing (prototype, 5–10 church volunteers).
  - Iterate based on feedback (e.g., simplify match table).
  - Handoff assets/specs to frontend team (Figma, GitHub).
- **Month 8**:
  - Support frontend development (React, Material-UI/Tailwind).
  - Conduct final user testing (React app, 10–15 volunteers).
  - Finalize accessibility audit and compliance report.
  - Deliver iteration recommendations for pilot (Month 8–9).

Total Effort: ~400–600 hours (~2–3 months full-time, within $20K–$40K budget).

.

9. Risk Mitigation
- **Non-Technical Users**:
  - Risk: Volunteers find match table or settings confusing.
  - Mitigation: Simplify layout (large buttons, clear labels), test with 10–15 volunteers, iterate based on feedback.

- **Performance**:
  - Risk: Slow UI updates (>200ms) impact <2s latency.
  - Mitigation: Optimize components (e.g., lazy-load history), verify with frontend developer.
- **Browser Compatibility**:
  - Risk: Chrome/Firefox differences break UI (e.g., Web Audio API).
  - Mitigation: Test on both browsers, provide Electron app as fallback.
- **Projector Variability**:

  - Risk: 720p/1080p projectors display text inconsistently.
  - Mitigation: Support resolution detection, test on multiple projectors during pilot.

Deliverables: Risk assessment, mitigation plan.

10. Success Criteria
- **Usability**: 90%+ of pilot users (10–15 volunteers) complete tasks (login, configure, select match, override) without assistance.
- **Speed**: <5s to select a match, <30s to start projection (user testing).
- **Satisfaction**: 80%+ rate UI as “easy to use” (5-point scale).
- **Accessibility**: 100% WCAG 2.1 Level AA compliance (audit report).
- **Adoption**: Pilot churches (5–10) adopt app for Sunday services (Month 8–9).

11. Resources
- **Design Tools**: Figma (primary), Adobe XD (optional), WAVE (accessibility).
- **Inspiration**:
  - ProPresenter, EasyWorship (church projection UIs).
  - Zoom (audio input selection).
  - Material-UI DataGrid (table design).
- **Standards**:
  - WCAG 2.1 Level AA (https://www.w3.org/TR/WCAG21/).
  - Material Design Guidelines (https://material.io/).

Conclusion
This guide equips the UI/UX Designer to create an intuitive, accessible, and efficient interface for the VerseProjection MVP, supporting continuous speech analysis, top 10 verse matching, user selection, and projection. The design prioritizes simplicity for non-technical church volunteers, with a clear Login Page, feature-rich Admin Dashboard, customizable Projection Window, and robust Error Notifications. The Electron app ensures offline functionality, reusing web assets for consistency. By adhering to WCAG 2.1, testing with

users, and collaborating with developers, the designer will deliver a UI that meets the PRD's requirements (<2s latency, 90%+ accuracy, scalability).