**Homepage Plan (Revised)**

The two-phase strategy is still perfect. The only change is where the button in Phase 1 directs the user.

* **Phase 1 (Now until Sept 30th): Driving Traffic to Unstop**
  + **Countdown Timer:** Correctly counts down to the **Registration Closing Date (25th September 2025)**. This creates urgency and reminds visitors that the deadline is approaching.
  + **Main Call-to-Action (CTA) Button:** The button text should be explicit. Instead of a generic "Register Now", it should be **"Register on Unstop"**.
    - **Action:** This button will link directly to your official WomenInDev 2.0 registration page on Unstop.
* **Phase 2 (From Oct 1st onwards): Focusing on Finalists**
  + **Countdown Timer:** Changes to count down to the **Final Round Start Date (9th October 2025, 6:00 PM IST)**.
  + **Main Call-to-Action (CTA) Button:** Changes to **"View Shortlisted Teams"**, linking to your /shortlisted-teams page.

**Clarifying the Participant Journey**

This makes the user flow very clear and distinct:

1. **Step 1: Registration (via Unstop):** A potential participant lands on your website. They explore the timeline, rules, and prizes. When they are ready, they click the "Register on Unstop" button, which takes them to the external Unstop page to complete the registration process.
2. **Step 2: Post-Shortlisting (Your Website):** Once the 15 finalist teams are selected and announced on **September 30th**, you will provide them with instructions to create an account (or provide them with credentials) to log in to **your website's dashboard**. This dashboard is an exclusive portal for finalists to access the problem statements and receive live updates during the hackathon.

This is an excellent and professional setup. The rest of the website plan, including the dynamic timeline, organizer details, and the post-login dashboard for finalists, remains perfectly structured for this workflow.

provide the entire work flow of the website and features

Show thinking

Of course. Here is a complete, consolidated workflow and feature breakdown for the WomenInDev 2.0 website. This document can serve as a blueprint for development.

**Complete Website Workflow and Feature Breakdown**

**I. High-Level Goal**

The website's primary purpose is to act as the central information and engagement hub for the WomenInDev 2.0 hackathon. It will attract participants, guide them to the registration platform (Unstop), and later transform into an exclusive portal for finalists to access critical event information.

**II. Key Personas (User Roles)**

1. **The Public Visitor (Potential Participant):** Any student visiting the site to learn about the event before the shortlisting is complete. Their goal is to find information and register.
2. **The Shortlisted Participant (Finalist):** A team member who has cleared the prelims. Their goal is to access exclusive content like problem statements and live announcements for the final round.
3. **The Administrator (Organizer):** A member of the organizing team. Their goal is to manage the website's dynamic content, such as releasing problem statements and posting announcements.

**III. Website Features Breakdown**

**A. Informational & Public-Facing Features**

These features are accessible to everyone.

* **Homepage:** The main landing page with a two-phase dynamic display.
* **About Page:** Detailed "Event Overview" and "Event Structure" sections.
* **Dynamic Timeline:** A visual representation of the event schedule that automatically updates to highlight past, current, and future stages.
* **Rules & Guidelines Page:** Clear, itemized list of eligibility, evaluation criteria, and general rules.
* **Prizes Page:** Information on the ₹40,000 prize pool and certificates.
* **Past Winners Gallery:** A "Hall of Fame" for WomenInDev 1.0, explicitly mentioning the Winner and Runners-Up with project details.
* **Organizers Page:** Profile cards for all four organizers (Mudita, Sidrah, Shreya, Aashi) with their photo, name, email, and phone number.
* **Community Link:** A prominent button/link directing users to the official **WhatsApp Group**.
* **Shortlisted Teams Page:** A dedicated page (activated after Sept 30th) listing the 15 finalist teams and 5 waitlisted teams.
* **FAQ Page:** A section to proactively answer common questions about teams, tech stacks, logistics, etc.

**B. Interactive & Authenticated Features**

These features require a login and are primarily for finalists and admins.

* **Secure Login System:** A single /login page that handles authentication. The system will differentiate between a regular user (finalist) and an admin based on their email credentials.
* **Personalized Dashboard:** The main portal for logged-in users. It will display a personalized welcome message (e.g., "Welcome, Team Cygnus!" or "Welcome, Mudita!").
* **Problem Statement Portal:** A section within the dashboard. It will initially show a "locked" or "coming soon" message. After an admin releases them, this page will display the detailed problem statements.
* **Live Announcement Bulletin:** A high-visibility banner or section on the user dashboard that displays real-time messages posted by admins.

**C. Administrator-Specific Features**

These are the controls available only to logged-in admins.

* **Admin Control Panel:** An exclusive tab or section on the dashboard, visible only to admins.
* **One-Click Problem Statement Release:** A single button within the admin panel (e.g., "Make Problems Live") that toggles the visibility of the problem statements for all finalists.
* **Announcement Publisher:** A simple text field and a "Post Announcement" button that allows admins to instantly update the message on the Live Announcement Bulletin for all users.

**IV. Complete User Workflows**

**Workflow A: The Journey of a Potential Participant (Now until Sept 30th)**

1. **Discovery:** A student hears about WomenInDev 2.0 and finds the website link.
2. **Arrival:** She lands on the homepage. The current date is **September 22, 2025**. She immediately sees a prominent **countdown timer ticking down to the registration deadline** on September 25th.
3. **Information Gathering:** She navigates through the site:
   * She reads the About page to understand the hackathon's vision.
   * She checks the Timeline page, which highlights that "Registration" is the current phase.
   * She reviews the Rules & Eligibility to confirm she can participate.
   * She is inspired by the projects on the Past Winners Gallery.
4. **Action:** Convinced, she clicks the main Call-to-Action button: **"Register on Unstop"**.
5. **Redirection:** She is taken to the official Unstop page where she completes her team's registration.
6. **Confirmation:** After the prelims, she revisits the site on September 30th. The homepage countdown now points to the Final Event, and the main button says **"View Shortlisted Teams"**. She clicks it and sees her team name on the list.

**Workflow B: The Journey of a Shortlisted Finalist (Oct 9th - 10th)**

1. **Onboarding:** The finalist receives an email with login instructions for the website's dashboard.
2. **Login:** She goes to the website's /login page and logs in with the provided credentials.
3. **Dashboard Access:** She is greeted with a personalized message ("Welcome, [Her Team Name]!"). At the top, she sees an announcement: "Check-in at the venue begins at 5:00 PM."
4. **Awaiting Problems:** She navigates to the "Problem Statements" tab, which displays a message: "Challenges will be revealed on 9th October 2025, 6:00 PM IST."
5. **Hackathon Start:** At exactly 6:00 PM on October 9th, she refreshes the page. The problem statements are now fully visible, with detailed descriptions and judging criteria.
6. **Staying Updated:** Throughout the 24-hour hackathon, she keeps the dashboard open to see live updates on the bulletin, like "Mentors are now available for frontend guidance" or "Reminder: 1 hour left for submission."
7. **Submission:** When her project is complete, she clicks the "Submit Your Project" button on the dashboard, which takes her directly to the submission Google Form.

**Workflow C: The Journey of an Administrator (During the Event)**

1. **Login:** Sidrah logs into the website with her admin email (bt23cse211@iiitn.ac.in).
2. **Admin View:** She lands on the dashboard, which looks similar to the user's but has an additional **"Admin Controls"** tab.
3. **Release Statements:** At 5:59 PM on October 9th, she navigates to the Admin Controls. At exactly 6:00 PM, she clicks the **"Release Problem Statements"** button. The website backend updates the state, and the problem statements become visible to all logged-in finalists.
4. **Post Announcement:** Later that evening, she needs to inform teams about dinner. She goes to the Admin Controls, types "Dinner is now served in the cafeteria. Enjoy your break!" into the announcement text box, and clicks **"Post Announcement"**. The message immediately appears on the dashboard of every finalist.
5. **Final Reminder:** The next day, she posts a final reminder: "Final submissions are due at 6:00 PM sharp. Please ensure your GitHub repo is public."

**Product & Vision Document**

**WomenInDev 2.0 Hackathon Platform**

**Executive Summary**

WomenInDev 2.0 is a comprehensive web platform designed to serve as the central hub for a women-focused hackathon event. The platform facilitates the complete participant journey from initial discovery through final event participation, employing a dual-phase strategy that evolves from public marketing to exclusive finalist portal.

**Vision Statement**

To create an intuitive, dynamic platform that seamlessly guides potential participants through hackathon discovery and registration while transforming into a powerful exclusive portal for finalists, ensuring optimal engagement and communication throughout the WomenInDev 2.0 experience.

**Product Goals**

**Primary Objectives**

* **Phase 1 Goal**: Drive maximum qualified registrations to the Unstop platform before the September 25th deadline
* **Phase 2 Goal**: Provide finalists with an exclusive, real-time communication and resource access portal

**Success Metrics**

* Registration conversion rate from website visits
* User engagement time on informational pages
* Finalist portal adoption and usage rates
* Real-time announcement delivery success

**Target Audience**

**Primary Personas**

**1. The Potential Participant**

* Demographics: Female students interested in technology
* Goals: Learn about the hackathon, understand requirements, complete registration
* Pain Points: Information scattered across platforms, unclear registration process
* Behavior: Research-oriented, seeks comprehensive information before committing

**2. The Shortlisted Finalist**

* Demographics: Technically skilled students who cleared preliminary rounds
* Goals: Access exclusive content, stay updated during the event, submit projects
* Pain Points: Missing critical updates, difficulty accessing real-time information
* Behavior: Task-focused, requires immediate access to current information

**3. The Event Administrator**

* Demographics: Hackathon organizers (Mudita, Sidrah, Shreya, Aashi)
* Goals: Manage event communication, control information release timing, monitor participant engagement
* Pain Points: Manual update processes, coordination across team members
* Behavior: Time-sensitive operations, need for instant global updates

**Market Context**

The hackathon landscape is highly competitive, with participants often juggling multiple events. WomenInDev 2.0 differentiates itself through:

* Focus on women in technology
* Structured two-phase participant journey
* Seamless integration between public information and exclusive finalist experience
* Real-time communication capabilities

**Product Positioning**

WomenInDev 2.0 positions itself as a premium, professionally managed hackathon with emphasis on:

* **Transparency**: Clear timelines, rules, and processes
* **Exclusivity**: Elite finalist experience with dedicated portal
* **Communication**: Real-time updates and announcements
* **Legacy**: Building on WomenInDev 1.0 success

**Key Value Propositions**

**For Participants**

* Single source of truth for all event information
* Streamlined registration process with clear call-to-actions
* Exclusive finalist experience with real-time updates
* Professional event management and communication

**For Organizers**

* Automated phase transitions based on dates
* Centralized announcement system
* Real-time control over information release
* Professional brand representation

**Competitive Analysis**

Traditional hackathon websites typically fall into two categories:

1. **Static Information Sites**: Provide basic details but lack dynamic engagement
2. **Complex Platforms**: Over-engineered solutions that confuse participants

WomenInDev 2.0 strikes the optimal balance with:

* Dynamic, date-aware content presentation
* Phase-appropriate functionality
* Intuitive user experience design
* Focused feature set avoiding feature bloat

**Product Strategy**

**Phase 1 Strategy (Until Sept 30th)**

* **Primary Focus**: Conversion optimization for Unstop registration
* **Key Features**: Countdown urgency, comprehensive information architecture
* **Success Measure**: Registration numbers on Unstop platform

**Phase 2 Strategy (Oct 1st onwards)**

* **Primary Focus**: Finalist engagement and support
* **Key Features**: Exclusive portal, real-time communication, resource access
* **Success Measure**: Finalist satisfaction and event execution quality

**Risk Assessment**

**Technical Risks**

* **Server capacity during high-traffic periods**
  + Mitigation: Scalable hosting infrastructure
* **Authentication system security**
  + Mitigation: Robust security protocols and testing

**Business Risks**

* **Low registration conversion**
  + Mitigation: Clear value proposition and streamlined user experience
* **Finalist portal adoption issues**
  + Mitigation: Comprehensive onboarding and intuitive design

**Future Roadmap**

Post-WomenInDev 2.0 considerations:

* Analytics dashboard for organizer insights
* Multi-event platform capabilities
* Enhanced community features
* Mobile application development
* Integration with additional registration platforms

**Stakeholder Alignment**

All four organizers are aligned on the dual-phase strategy and core functionality requirements. The platform design reflects their collective vision for a professional, engaging hackathon experience that honors the WomenInDev legacy while introducing innovative participant engagement strategies.

**Functional & Feature Documentation**

**WomenInDev 2.0 Platform**

**Feature Categories Overview**

The platform is organized into three distinct feature categories based on user access levels and functionality requirements.

**A. Public-Facing Features (No Authentication Required)**

**1. Dynamic Homepage**

**Functionality**: Date-aware content presentation with phase-specific call-to-actions

**Phase 1 Behavior (Until Sept 30th)**:

* Displays countdown timer to Registration Closing Date (Sept 25th, 2025)
* Primary CTA button: "Register on Unstop" (links to external Unstop registration)
* Hero section emphasizes registration urgency
* Quick access to event information

**Phase 2 Behavior (From Oct 1st)**:

* Countdown timer switches to Final Round Start Date (Oct 9th, 6:00 PM IST)
* Primary CTA button: "View Shortlisted Teams" (links to /shortlisted-teams page)
* Hero section emphasizes finalist selection and upcoming event

**Technical Requirements**:

* Real-time date comparison logic
* Automatic content switching at midnight Oct 1st
* Responsive countdown timer with precise timezone handling (IST)

**2. About Page**

**Functionality**: Comprehensive event information presentation

**Content Sections**:

* Event Overview: Mission, vision, and objectives of WomenInDev 2.0
* Event Structure: Detailed breakdown of preliminary and final rounds
* Hackathon Timeline: Key dates and milestones
* Participant Benefits: What participants gain from participation

**Technical Requirements**:

* Static content with occasional manual updates
* SEO-optimized content structure
* Mobile-responsive layout

**3. Dynamic Timeline**

**Functionality**: Visual progress tracker showing event stages

**Timeline Stages**:

1. Registration Open (Aug 15 - Sept 25)
2. Preliminary Evaluation (Sept 26 - 29)
3. Shortlisting Announcement (Sept 30)
4. Final Event (Oct 9-10)
5. Results & Awards (Oct 10)

**Visual Indicators**:

* Completed stages: Green checkmark
* Current stage: Animated highlight
* Future stages: Gray/pending state

**Technical Requirements**:

* Automatic stage progression based on current date
* Smooth animations for state transitions
* Progress bar calculation

**4. Rules & Guidelines**

**Functionality**: Detailed participant requirements and evaluation criteria

**Content Sections**:

* Eligibility Requirements
* Team Formation Guidelines
* Technology Stack Rules
* Submission Requirements
* Evaluation Criteria (Innovation, Technical Implementation, Impact, Presentation)
* Code of Conduct
* Disqualification Conditions

**Technical Requirements**:

* Searchable content
* Anchor links for easy navigation
* Downloadable PDF version

**5. Prizes Information**

**Functionality**: Prize pool breakdown and certificate details

**Content**:

* Total Prize Pool: ₹40,000
* Prize Distribution: Winner, First Runner-up, Second Runner-up
* Certificate Details: All participants receive participation certificates
* Special Recognition Categories

**6. Past Winners Gallery**

**Functionality**: WomenInDev 1.0 success stories and project showcase

**Content Structure**:

* Winner Project: Title, team members, project description, technology stack
* Runner-up Projects: Similar detailed presentations
* Event Statistics: Participation numbers, success metrics
* Testimonials: Past participant experiences

**Technical Requirements**:

* Image optimization for fast loading
* Project detail modal/expansion functionality

**7. Organizers Page**

**Functionality**: Transparent team presentation building trust and accessibility

**Organizer Profiles** (4 total):

* Mudita Singh
* Sidrah Fatima (bt23cse211@iiitn.ac.in)
* Shreya Sharma
* Aashi Agarwal

**Information per Organizer**:

* Professional photograph
* Name and role
* Contact email
* Phone number
* Brief bio/background

**Technical Requirements**:

* Responsive card layout
* Contact information click-to-call/email functionality

**8. Community Integration**

**Functionality**: Direct connection to official communication channels

**Features**:

* WhatsApp Group join button
* Social media links
* Community guidelines preview

**9. Shortlisted Teams Page**

**Functionality**: Post-preliminary results presentation

**Activation**: Available from Oct 1st onwards

**Content**:

* 15 Finalist Teams: Team names and brief descriptions
* 5 Waitlisted Teams: With clear waitlist explanation
* Next steps for finalists
* Login instructions for dashboard access

**Technical Requirements**:

* Date-based content activation
* Team data management system
* Clear visual distinction between finalists and waitlisted teams

**10. FAQ Section**

**Functionality**: Proactive question answering to reduce support burden

**Question Categories**:

* Registration Process
* Team Formation
* Technical Requirements
* Event Logistics
* Submission Process
* Judging Criteria

**B. Authenticated Features (Login Required)**

**1. Secure Authentication System**

**Functionality**: Role-based access control for finalists and administrators

**User Types**:

* Finalist Teams: Individual login credentials per team
* Administrators: Special privileges using organizer emails

**Security Features**:

* Encrypted password storage
* Session management
* Role-based permission system
* Secure logout functionality

**Technical Requirements**:

* JWT or session-based authentication
* Password reset functionality
* Account lockout after failed attempts

**2. Personalized Dashboard**

**Functionality**: Central hub for authenticated users with role-appropriate content

**Common Elements**:

* Personalized welcome message ("Welcome, Team [Name]!" or "Welcome, [Admin Name]!")
* Navigation menu with role-appropriate options
* Real-time announcement display
* Current event status

**Finalist-Specific Elements**:

* Team information summary
* Problem statement access
* Submission portal link
* Event schedule and updates

**Admin-Specific Elements**:

* Admin control panel access
* System status overview
* User activity monitoring

**3. Problem Statement Portal**

**Functionality**: Controlled access to hackathon challenges

**Pre-Release State**:

* "Coming Soon" message with exact release time
* Countdown to statement release (Oct 9th, 6:00 PM IST)
* Placeholder content maintaining engagement

**Post-Release State**:

* Complete problem statements with detailed descriptions
* Evaluation criteria for each problem
* Resource links and documentation
* FAQ for problem-specific questions

**Technical Requirements**:

* Admin-controlled release mechanism
* Real-time content updates
* Mobile-optimized reading experience

**4. Live Announcement System**

**Functionality**: Real-time communication from organizers to all users

**Display Features**:

* Prominent banner on dashboard
* Timestamp for each announcement
* Message priority levels (normal, important, urgent)
* Read/unread indicators

**User Interaction**:

* Mark as read functionality
* Announcement history access
* Mobile push notification capability (future enhancement)

**C. Administrator-Exclusive Features**

**1. Admin Control Panel**

**Functionality**: Centralized management interface for event control

**Access Control**:

* Restricted to organizer email addresses
* Secondary authentication for sensitive operations
* Activity logging for all admin actions

**2. Problem Statement Release Control**

**Functionality**: Precise timing control for critical event moments

**Features**:

* Single-click release mechanism
* Confirmation dialog to prevent accidental releases
* Automatic timestamp logging
* Rollback capability (emergency use only)

**Technical Requirements**:

* Database flag system for content visibility
* Real-time update propagation to all user sessions
* Audit trail maintenance

**3. Announcement Management System**

**Functionality**: Instant communication tool for event coordination

**Features**:

* Rich text message composition
* Priority level selection
* Target audience selection (all users, finalists only, specific teams)
* Message scheduling for future delivery
* Message edit/delete capabilities (within time limits)

**Technical Requirements**:

* Real-time message delivery system
* Message persistence in database
* WebSocket or similar for instant updates

**4. User Management Interface**

**Functionality**: Oversight and control of user accounts and access

**Features**:

* User account creation/deactivation
* Password reset initiation
* Session monitoring and termination
* Activity log review
* Bulk operations for team management

**Feature Integration & Dependencies**

**Cross-Feature Dependencies**

1. **Authentication ↔ All Authenticated Features**: Core dependency for access control
2. **Date System ↔ Dynamic Content**: Timeline, homepage, and announcement features depend on accurate date handling
3. **Admin Controls ↔ User Experience**: Admin actions directly impact finalist user interface

**Performance Requirements**

* **Page Load Time**: < 3 seconds for all pages
* **Real-time Updates**: < 2 seconds for announcement delivery
* **Concurrent Users**: Support for 100+ simultaneous users during peak times
* **Mobile Responsiveness**: Full functionality on mobile devices

**Accessibility Requirements**

* WCAG 2.1 AA compliance
* Screen reader compatibility
* Keyboard navigation support
* High contrast mode support
* Font size scalability

**Browser Compatibility**

* Chrome 90+
* Firefox 85+
* Safari 14+
* Edge 90+
* Mobile browsers (iOS Safari, Chrome Mobile)

**Design Documentation**

**WomenInDev 2.0 Platform**

**Design Philosophy**

The WomenInDev 2.0 platform embraces a modern, professional aesthetic that balances technical sophistication with accessibility. The design philosophy centers on three core principles:

1. **Progressive Disclosure**: Information is revealed appropriately based on user context and event phase
2. **Empowerment Through Clarity**: Clean, confident design that reflects the strength and capability of women in technology
3. **Dynamic Responsiveness**: Visual elements that adapt not just to screen size, but to event timeline and user status

**Visual Identity**

**Color Palette**

**Primary Colors**:

* **Deep Purple** (#6B46C1): Primary brand color representing innovation and technology
* **Vibrant Pink** (#EC4899): Secondary accent color symbolizing empowerment and energy
* **Electric Blue** (#3B82F6): Supporting color for interactive elements and progress indicators

**Neutral Colors**:

* **Charcoal** (#1F2937): Primary text and high-contrast elements
* **Warm Gray** (#6B7280): Secondary text and borders
* **Light Gray** (#F3F4F6): Background sections and card backgrounds
* **Pure White** (#FFFFFF): Primary background and card surfaces

**Semantic Colors**:

* **Success Green** (#10B981): Completed timeline stages, success messages
* **Warning Amber** (#F59E0B): Urgent announcements, countdown timers
* **Error Red** (#EF4444): Error states, critical warnings

**Typography**

**Primary Typeface**: Inter

* Modern, highly legible sans-serif
* Excellent readability across all screen sizes
* Strong geometric foundation with humanist touches

**Heading Hierarchy**:

* **H1** (36px/40px): Page titles, hero headings
* **H2** (30px/36px): Section headers
* **H3** (24px/32px): Subsection headers
* **H4** (20px/28px): Card titles, form labels
* **Body** (16px/24px): Primary text content
* **Small** (14px/20px): Supporting text, metadata

**Font Weights**:

* **Regular (400)**: Body text, standard content
* **Medium (500)**: Emphasized text, navigation
* **Semibold (600)**: Section headers, button text
* **Bold (700)**: Primary headings, hero text

**Logo and Branding**

**Logo Treatment**:

* Clean, modern wordmark "WomenInDev 2.0"
* Gradient application using primary color palette
* Scalable SVG format for crisp rendering at all sizes
* Monochrome version for single-color applications

**Brand Voice**:

* Professional yet approachable
* Confident and empowering
* Clear and direct communication
* Inclusive and welcoming tone

**Layout Systems**

**Grid Structure**

**Desktop Layout** (1200px+ containers):

* 12-column CSS Grid system
* 24px gutters between columns
* Maximum content width: 1200px
* Centered content with equal margins

**Tablet Layout** (768px - 1199px):

* 8-column grid adaptation
* 20px gutters
* Full-width content with 20px side margins

**Mobile Layout** (< 768px):

* Single column layout
* 16px side margins
* Stack-first approach for all components

**Spacing System**

**Base Unit**: 8px

**Spacing Scale**:

* **XS**: 4px (0.5 units)
* **SM**: 8px (1 unit)
* **MD**: 16px (2 units)
* **LG**: 24px (3 units)
* **XL**: 32px (4 units)
* **2XL**: 48px (6 units)
* **3XL**: 64px (8 units)

**Component Design Specifications**

**Navigation Header**

**Structure**:

* Fixed header with transparent background becoming solid on scroll
* Logo positioned left
* Main navigation center-aligned
* CTA button right-aligned

**States**:

* **Default**: Transparent background with shadow on scroll
* **Mobile**: Hamburger menu with slide-out navigation
* **Authenticated**: Additional user avatar and dropdown

**Responsive Behavior**:

* Desktop: Horizontal navigation with all items visible
* Tablet: Condensed navigation with abbreviated labels
* Mobile: Hamburger menu with full-screen overlay

**Hero Section**

**Phase 1 Design**:

* Large countdown timer as focal point
* Gradient background with subtle geometric patterns
* "Register on Unstop" button with prominent styling
* Supporting text emphasizing urgency and opportunity

**Phase 2 Design**:

* Countdown to final event
* Celebration graphics for shortlisted status
* "View Shortlisted Teams" primary action
* Achievement-focused messaging

**Technical Specifications**:

* Minimum height: 60vh on desktop, 80vh on mobile
* Background: Animated gradient with CSS keyframes
* Countdown: Large, bold numerics with descriptive labels

**Timeline Component**

**Visual Design**:

* Horizontal progress bar on desktop
* Vertical timeline on mobile
* Interactive hover states revealing additional details
* Smooth transitions between timeline states

**Stage Indicators**:

* **Completed**: Filled circle with checkmark icon
* **Current**: Pulsing animated circle with primary color
* **Upcoming**: Outlined circle with muted colors

**Content Cards**:

* Each stage expands to show detailed information
* Card design with subtle shadows and rounded corners
* Consistent spacing and typography hierarchy

**Authentication Interface**

**Login Form Design**:

* Centered modal or dedicated page layout
* Clean, minimal form design
* Clear error state messaging
* Social login options (future enhancement)

**Dashboard Layout**:

* Sidebar navigation on desktop
* Top tab navigation on mobile
* Main content area with card-based information architecture
* Persistent announcement banner

**Card Components**

**Standard Card**:

* White background with subtle shadow
* 12px border radius
* 24px internal padding
* Hover elevation increase

**Information Cards**:

* Icon + title + description layout
* Consistent aspect ratios
* Color-coded categories

**Team Profile Cards**:

* Photo, name, contact information
* Hover effects revealing additional details
* Consistent sizing across all profiles

**Button System**

**Primary Buttons**:

* Gradient background using brand colors
* White text with medium font weight
* 12px border radius
* 16px vertical, 24px horizontal padding
* Hover state: Slight scale increase + shadow

**Secondary Buttons**:

* Outlined style with primary color border
* Primary color text
* Same sizing as primary buttons
* Hover state: Background fill with color inversion

**Utility Buttons**:

* Smaller size for secondary actions
* Subtle styling to avoid competing with primary actions

**Form Elements**

**Input Fields**:

* Clean, modern styling with subtle borders
* Focus states with primary color highlights
* Error states with red border and message
* Consistent height: 48px
* 12px border radius

**Validation**:

* Real-time validation with inline error messages
* Success states with green highlights
* Clear, actionable error messaging

**Animation and Interaction Design**

**Micro-Interactions**

**Hover Effects**:

* Subtle scale transforms (102-105% scale)
* Color transitions over 200ms
* Shadow elevation changes
* Icon animations for interactive elements

**Loading States**:

* Skeleton loading for content areas
* Spinner animations for form submissions
* Progress indicators for multi-step processes

**Transitions**:

* Page transitions: 300ms ease-in-out
* Component state changes: 200ms ease
* Modal/overlay animations: 250ms ease-out

**Countdown Timer Animation**

* Smooth number transitions with flip animation
* Color changes as deadline approaches
* Subtle pulsing effect during critical timeframes

**Real-time Updates**

* Gentle slide-in animation for new announcements
* Subtle highlight flash for updated content
* Non-intrusive notification badges

**Responsive Design Strategy**

**Breakpoint System**

**Mobile First Approach**:

* **XS**: 0px - 575px (Small phones)
* **SM**: 576px - 767px (Large phones)
* **MD**: 768px - 991px (Tablets)
* **LG**: 992px - 1199px (Small desktops)
* **XL**: 1200px+ (Large desktops)

**Component Adaptations**

**Navigation**:

* Desktop: Horizontal menu bar
* Mobile: Collapsible hamburger menu

**Cards**:

* Desktop: Multi-column grid layouts
* Mobile: Single column stack

**Forms**:

* Desktop: Multi-column layouts where appropriate
* Mobile: Single column with larger touch targets

**Timeline**:

* Desktop: Horizontal progress visualization
* Mobile: Vertical timeline with expanded touch areas

**Accessibility Design**

**Color Accessibility**

* All color combinations meet WCAG AA contrast requirements
* Color is never the sole method of conveying information
* High contrast mode compatibility

**Typography Accessibility**

* Minimum text size: 16px for body text
* Line height: 1.5x font size minimum
* Character spacing optimized for readability

**Interactive Element Accessibility**

* All interactive elements minimum 44px touch target
* Clear focus indicators for keyboard navigation
* Screen reader-friendly alt text and descriptions

**Motion Accessibility**

* Respect prefers-reduced-motion CSS media query
* Alternative static indicators for animated elements
* Optional motion disable control

**Design System Documentation**

**Component Library Structure**

Each component includes:

* Visual specifications and measurements
* Code implementation guidelines
* Usage examples and best practices
* Accessibility requirements
* Responsive behavior documentation

**Design Token System**

Centralized design tokens for:

* Color values and semantic naming
* Typography scales and weights
* Spacing and sizing values
* Border radius and shadow specifications
* Animation timing and easing values

**Maintenance Guidelines**

* Regular design review schedule
* User feedback integration process
* Performance impact assessment for design changes
* Cross-browser compatibility testing requirements

**Brand Implementation**

**Photography Style**

* High-quality, professional images of diverse women in technology
* Bright, energetic lighting
* Modern workplace and technology environments
* Authentic, candid moments over posed portraits

**Illustration Style**

* Modern, geometric illustrations
* Primary color palette integration
* Technology and empowerment themes
* Scalable vector format for all screen densities

**Voice and Tone**

* **Professional**: Competent and knowledgeable
* **Empowering**: Confidence-building and supportive
* **Inclusive**: Welcoming to all skill levels and backgrounds
* **Clear**: Direct communication without jargon

This design system ensures consistency across all platform touchpoints while maintaining the flexibility to evolve with user needs and technological improvements.

**Technical Documentation**

**WomenInDev 2.0 Platform**

**System Architecture Overview**

The WomenInDev 2.0 platform follows a modern, scalable architecture designed for high performance and maintainability. The system is built using a separation of concerns approach with distinct frontend and backend layers.

**Technology Stack**

**Frontend Technologies**

**Framework**: React 18+ with Next.js 13+

* **Rationale**: Server-side rendering for SEO optimization, built-in routing, and excellent performance
* **State Management**: React Context + useReducer for complex state, useState for local component state
* **CSS Framework**: Tailwind CSS 3.0+ for utility-first styling
* **Type Safety**: TypeScript for enhanced developer experience and reduced runtime errors

**Additional Frontend Libraries**:

* **date-fns**: Robust date manipulation and timezone handling
* **framer-motion**: Smooth animations and transitions
* **react-hook-form**: Form state management and validation
* **zod**: Schema validation for forms and API responses
* **lucide-react**: Modern icon library
* **recharts**: Data visualization for analytics dashboard

**Backend Technologies**

**Runtime**: Node.js 18+ LTS **Framework**: Express.js with TypeScript

* **Database**: PostgreSQL 14+ for relational data with Redis for caching/sessions
* **ORM**: Prisma for type-safe database operations
* **Authentication**: JWT with refresh token strategy
* **File Storage**: AWS S3 or similar for static assets
* **Email Service**: SendGrid or AWS SES for transactional emails

**Backend Libraries**:

* **bcryptjs**: Password hashing
* **helmet**: Security headers
* **cors**: Cross-origin resource sharing
* **express-rate-limit**: API rate limiting
* **winston**: Logging framework
* **joi**: Server-side validation

**Infrastructure & Deployment**

**Hosting**: Vercel (Frontend) + Railway/DigitalOcean (Backend) **CDN**: Cloudflare for global content delivery **Monitoring**: Sentry for error tracking **Analytics**: Google Analytics 4 + custom event tracking

**Database Design**

**Entity Relationship Model**

**Users Table**:

CREATE TABLE users (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

email VARCHAR(255) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

role ENUM('admin', 'participant') DEFAULT 'participant',

team\_name VARCHAR(255),

team\_members JSONB,

is\_shortlisted BOOLEAN DEFAULT FALSE,

is\_waitlisted BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Announcements Table**:

CREATE TABLE announcements (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

title VARCHAR(255) NOT NULL,

content TEXT NOT NULL,

priority ENUM('normal', 'important', 'urgent') DEFAULT 'normal',

target\_audience ENUM('all', 'finalists', 'specific') DEFAULT 'all',

created\_by UUID REFERENCES users(id),

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**System\_Settings Table**:

CREATE TABLE system\_settings (

key VARCHAR(255) PRIMARY KEY,

value TEXT NOT NULL,

updated\_by UUID REFERENCES users(id),

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**User\_Sessions Table**:

CREATE TABLE user\_sessions (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

user\_id UUID REFERENCES users(id),

token\_hash VARCHAR(255) NOT NULL,

expires\_at TIMESTAMP NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Database Optimization**

* **Indexing Strategy**: Composite indexes on frequently queried columns (email, role, is\_shortlisted)
* **Connection Pooling**: Maximum 20 connections for production environment
* **Query Optimization**: Prepared statements and query analysis
* **Backup Strategy**: Daily automated backups with 30-day retention

**API Design Specification**

**RESTful API Endpoints**

**Authentication Endpoints**:

POST /api/auth/login

Body: { email: string, password: string }

Response: { token: string, user: UserObject, expiresIn: number }

POST /api/auth/refresh

Body: { refreshToken: string }

Response: { token: string, expiresIn: number }

POST /api/auth/logout

Headers: { Authorization: "Bearer <token>" }

Response: { message: string }

**User Management Endpoints**:

GET /api/users/profile

Headers: { Authorization: "Bearer <token>" }

Response: { user: UserObject }

PUT /api/users/profile

Headers: { Authorization: "Bearer <token>" }

Body: { teamName?: string, teamMembers?: Array<string> }

Response: { user: UserObject }

POST /api/users/create (Admin only)

Body: { email: string, password: string, teamName: string, role: string }

Response: { user: UserObject }

**Announcement Endpoints**:

GET /api/announcements

Headers: { Authorization: "Bearer <token>" }

Query: { page?: number, limit?: number }

Response: { announcements: Array<AnnouncementObject>, total: number }

POST /api/announcements (Admin only)

Body: { title: string, content: string, priority: string, targetAudience: string }

Response: { announcement: AnnouncementObject }

PUT /api/announcements/:id (Admin only)

Body: { title?: string, content?: string, priority?: string }

Response: { announcement: AnnouncementObject }

**System Control Endpoints**:

GET /api/system/settings

Response: { problemStatementsReleased: boolean, currentPhase: string }

PUT /api/system/release-problems (Admin only)

Response: { success: boolean, releasedAt: timestamp }

GET /api/teams/shortlisted

Response: { finalists: Array<TeamObject>, waitlisted: Array<TeamObject> }

**API Security Implementation**

**Authentication Strategy**:

* JWT tokens with 15-minute expiration
* Refresh tokens with 7-day expiration
* Secure HTTP-only cookies for refresh tokens
* Role-based access control (RBAC)

**Request Validation**:

* Input sanitization using Joi schemas
* Rate limiting: 100 requests per 15 minutes per IP
* CORS configured for specific domains only
* Helmet.js for security headers

**Error Handling**:

interface APIError {

success: false;

error: {

code: string;

message: string;

details?: any;

};

timestamp: string;

}

**Frontend Architecture**

**Component Structure**

src/

├── components/ # Reusable UI components

│ ├── ui/ # Basic UI elements (buttons, inputs)

│ ├── forms/ # Form components

│ ├── layout/ # Layout components (header, footer)

│ └── features/ # Feature-specific components

├── pages/ # Next.js pages

├── hooks/ # Custom React hooks

├── contexts/ # React Context providers

├── utils/ # Utility functions and helpers

├── types/ # TypeScript type definitions

├── styles/ # Global styles and Tailwind config

└── lib/ # Third-party integrations and configs

**State Management Strategy**

**Global State (React Context)**:

interface AppState {

user: User | null;

isAuthenticated: boolean;

currentPhase: 'registration' | 'shortlisting' | 'finals';

announcements: Announcement[];

systemSettings: SystemSettings;

}

interface User {

id: string;

email: string;

role: 'admin' | 'participant';

teamName?: string;

teamMembers?: string[];

isShortlisted: boolean;

isWaitlisted: boolean;

}

**Local Component State**:

* Form state using react-hook-form
* UI state (modals, loading states) using useState
* Animation states using framer-motion

**Custom Hooks Implementation**

**useAuth Hook**:

const useAuth = () => {

const [user, setUser] = useState<User | null>(null);

const [loading, setLoading] = useState(true);

const login = async (email: string, password: string) => {

// Implementation

};

const logout = async () => {

// Implementation

};

return { user, login, logout, loading };

};

**useCountdown Hook**:

const useCountdown = (targetDate: Date) => {

const [timeLeft, setTimeLeft] = useState<TimeLeft>({

days: 0, hours: 0, minutes: 0, seconds: 0

});

useEffect(() => {

// Countdown logic with cleanup

}, [targetDate]);

return timeLeft;

};

**useRealTimeAnnouncements Hook**:

const useRealTimeAnnouncements = () => {

const [announcements, setAnnouncements] = useState<Announcement[]>([]);

useEffect(() => {

// WebSocket or polling implementation

// Auto-refresh every 30 seconds for new announcements

}, []);

return announcements;

};

**Performance Optimization**

**Frontend Performance**

**Code Splitting Strategy**:

* Route-based splitting using Next.js dynamic imports
* Component-based splitting for heavy components
* Third-party library splitting

**Image Optimization**:

* Next.js Image component for automatic optimization
* WebP format with fallbacks
* Lazy loading for below-the-fold images
* Responsive image sizing

**Caching Strategy**:

* Static assets: 1 year cache with versioning
* API responses: 5-minute cache for non-critical data
* User-specific data: No cache or very short TTL

**Bundle Optimization**:

* Tree shaking for unused code elimination
* Compression using gzip/brotli
* Critical CSS inlining
* Preloading of critical resources

**Backend Performance**

**Database Optimization**:

* Connection pooling (max 20 connections)
* Query optimization with EXPLAIN ANALYZE
* Proper indexing strategy
* Database query caching with Redis

**API Optimization**:

* Response compression
* Pagination for list endpoints
* Field selection to reduce payload size
* Request/response caching where appropriate

**Real-time Updates**:

// WebSocket implementation for live announcements

const WebSocketManager = {

connections: new Map(),

broadcast: (message: any, targetRole?: string) => {

this.connections.forEach((ws, userId) => {

if (!targetRole || user.role === targetRole) {

ws.send(JSON.stringify(message));

}

});

}

};

**Security Implementation**

**Authentication & Authorization**

**JWT Implementation**:

interface JWTPayload {

userId: string;

email: string;

role: 'admin' | 'participant';

iat: number;

exp: number;

}

const generateTokens = (user: User) => {

const accessToken = jwt.sign(

{ userId: user.id, email: user.email, role: user.role },

process.env.JWT\_SECRET!,

{ expiresIn: '15m' }

);

const refreshToken = jwt.sign(

{ userId: user.id },

process.env.REFRESH\_SECRET!,

{ expiresIn: '7d' }

);

return { accessToken, refreshToken };

};

**Password Security**:

* bcrypt with salt rounds: 12
* Password strength requirements enforced
* Account lockout after 5 failed attempts
* Password reset with secure tokens

**Request Security**:

* Input validation and sanitization
* SQL injection prevention through parameterized queries
* XSS protection with Content Security Policy
* CSRF protection for state-changing operations

**Data Protection**

**Sensitive Data Handling**:

* Environment variables for secrets
* Database encryption for sensitive fields
* Secure session management
* HTTPS enforcement in production

**Privacy Compliance**:

* Minimal data collection principle
* Clear data retention policies
* User consent mechanisms
* Data deletion capabilities

**Deployment & DevOps**

**CI/CD Pipeline**

**GitHub Actions Workflow**:

name: Deploy WomenInDev 2.0

on:

push:

branches: [main]

pull\_request:

branches: [main]

jobs:

test:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3

- name: Setup Node.js

uses: actions/setup-node@v3

with:

node-version: '18'

- name: Install dependencies

run: npm ci

- name: Run tests

run: npm test

- name: Run linting

run: npm run lint

- name: Type checking

run: npm run type-check

**Deployment Strategy**:

* Frontend: Vercel with automatic deployments from main branch
* Backend: Railway/DigitalOcean with Docker containerization
* Database: Managed PostgreSQL service
* Staging environment for pre-production testing

**Environment Configuration**

**Environment Variables**:

# Database

DATABASE\_URL=postgresql://...

REDIS\_URL=redis://...

# Authentication

JWT\_SECRET=...

REFRESH\_SECRET=...

# External Services

SENDGRID\_API\_KEY=...

AWS\_ACCESS\_KEY\_ID=...

AWS\_SECRET\_ACCESS\_KEY=...

# Application

NODE\_ENV=production

FRONTEND\_URL=https://womenindev2024.com

BACKEND\_URL=https://api.womenindev2024.com

**Monitoring & Logging**

**Application Monitoring**:

* Sentry for error tracking and performance monitoring
* Custom metrics for business logic (registrations, logins)
* Uptime monitoring with alerts
* Database performance monitoring

**Logging Strategy**:

import winston from 'winston';

const logger = winston.createLogger({

level: 'info',

format: winston.format.combine(

winston.format.timestamp(),

winston.format.json()

),

transports: [

new winston.transports.File({ filename: 'error.log', level: 'error' }),

new winston.transports.File({ filename: 'combined.log' })

]

});

// Usage example

logger.info('User login attempt', { email: user.email, ip: req.ip });

**Testing Strategy**

**Frontend Testing**

**Unit Testing**:

* Jest + React Testing Library
* Component testing with mock dependencies
* Custom hook testing
* Utility function testing

**Integration Testing**:

* API integration tests
* Form submission flows
* Authentication flows
* Real-time update functionality

**E2E Testing**:

* Playwright for critical user journeys
* Registration flow testing
* Admin functionality testing
* Cross-browser compatibility testing

**Backend Testing**

**Unit Testing**:

* Jest for business logic
* Database model testing
* API endpoint testing with supertest
* Authentication middleware testing

**Integration Testing**:

* Database integration tests
* Third-party service integration
* Email service testing
* File upload testing

**Load Testing**:

* Artillery.js for API load testing
* Database performance under load
* Concurrent user simulation
* Peak traffic handling (registration deadline)

**Maintenance & Scalability**

**Code Quality Standards**

**Linting & Formatting**:

* ESLint with TypeScript rules
* Prettier for consistent formatting
* Husky for pre-commit hooks
* Conventional commit messages

**Code Review Process**:

* All changes require PR review
* Automated testing on PR creation
* Security scan integration
* Performance impact assessment

**Scalability Considerations**

**Horizontal Scaling**:

* Stateless backend design
* Load balancer configuration
* Database read replicas for heavy read operations
* CDN for static asset delivery

**Performance Monitoring**:

* Response time tracking
* Database query performance
* Memory usage monitoring
* Error rate tracking

**Maintenance Schedule**

**Regular Maintenance Tasks**:

* Weekly dependency updates
* Monthly security patch reviews
* Quarterly performance optimization reviews
* Database maintenance and optimization

**Backup & Recovery**:

* Daily automated database backups
* Weekly full system backups
* Recovery procedure documentation
* Disaster recovery testing

**Documentation & Support**

**Developer Documentation**

**API Documentation**:

* OpenAPI/Swagger specification
* Interactive API explorer
* Code examples in multiple languages
* Authentication flow documentation

**Setup Instructions**:

# Local development setup

git clone https://github.com/womenindev/platform-v2.git

cd platform-v2

# Install dependencies

npm install

# Setup environment variables

cp .env.example .env.local

# Start development servers

npm run dev:frontend # Port 3000

npm run dev:backend # Port 8000

**Troubleshooting Guide**:

* Common development issues
* Database connection problems
* Authentication debugging
* Performance optimization tips

This technical documentation provides a comprehensive foundation for developing, deploying, and maintaining the WomenInDev 2.0 platform with enterprise-grade reliability and performance.