Mobile AI Interview Coach - Technical Development Roadmap

Bolt Hackathon 4-Week Implementation Strategy

Executive Summary

This roadmap outlines a comprehensive 4-week development strategy for building a Mobile AI Interview Coach app for the bolt hackathon, with a clear path to SaaS scaling. The app will leverage AI avatars (Tavus), voice synthesis (ElevenLabs), and subscription management (RevenueCat) to create an immersive interview preparation experience.

Target Timeline: 4 weeks (End of June 2024)
Primary Platform: bolt.new optimized web application with mobile-first design
Demo Date: End of June 2024

---

1. MVP Feature Prioritization & Scope Definition

Core MVP Features (Must-Have)

1. **AI Interview Simulator**
   - Text-based interview questions with AI responses
   - Basic question categories (behavioral, technical, industry-specific)
   - Simple scoring system based on response analysis

2. **User Authentication & Profiles**
   - Email/password registration
   - Basic profile setup (name, target role, experience level)
   - Interview history tracking

3. **Question Bank Management**
   - Pre-loaded common interview questions (50+ questions)
   - Category filtering (behavioral, technical, situational)
   - Difficulty levels (entry, mid, senior)

4. **Basic Analytics Dashboard**
   - Interview completion rates
   - Performance scores over time
   - Improvement suggestions

Enhanced Features (Should-Have)

1. **AI Avatar Integration (Tavus)**
   - Single AI interviewer avatar
   - Basic lip-sync with generated responses
   - Professional appearance customization

2. **Voice Interaction (ElevenLabs)**
   - Text-to-speech for interview questions
   - Voice response recording and analysis
   - Basic speech-to-text conversion

3. **Subscription System (RevenueCat)**
   - Free tier (5 interviews/month)
   - Premium tier ($9.99/month - unlimited interviews)
   - Basic payment processing

Future Features (Could-Have)

1. **Advanced AI Coaching**
   - Real-time feedback during interviews
   - Personalized improvement plans
   - Industry-specific interview prep

2. **Social Features**
   - Interview sharing with mentors
   - Community feedback system
   - Leaderboards and achievements

---

2. Technology Stack - Bolt.new Optimized

Frontend Framework

- **Next.js 14** with App Router
- **TypeScript** for type safety
- **Tailwind CSS** for rapid styling
- **Framer Motion** for smooth animations
- **React Hook Form** for form management

State Management

- **Zustand** (lightweight, bolt.new friendly)
- **TanStack Query** for server state management
- **Local Storage** for offline capabilities

UI Components

- **Shadcn/ui** components (bolt.new compatible)
- **Lucide React** for icons
- **Recharts** for analytics visualization
- **React Hot Toast** for notifications

Backend & Database

- **Supabase** (PostgreSQL + Auth + Storage)
- **Prisma** ORM for database management
- **Edge Functions** for serverless API endpoints

AI & Media Processing

- **OpenAI GPT-4** for interview question generation and analysis
- **Tavus API** for AI avatar generation
- **ElevenLabs API** for voice synthesis
- **Web Speech API** for voice recognition

Authentication & Payments

- **Supabase Auth** for user management
- **RevenueCat Web SDK** for subscription management
- **Stripe** integration via RevenueCat

Deployment & Hosting

- **Vercel** (optimal for bolt.new projects)
- **Supabase** for backend services
- **Cloudinary** for media asset management

---

3. Sponsor API Integration Requirements

3.1 Tavus Integration (AI Avatars)

API Endpoints:
- `POST /v2/replicas` - Create AI interviewer replica
- `POST /v2/videos` - Generate interview responses
- `GET /v2/videos/{video_id}` - Retrieve generated content

Implementation Strategy:

```
// Tavus SDK Integration
import { Tavus } from 'tavus-python-sdk';

const tavusClient = new Tavus({
  apiKey: process.env.TAVUS_API_KEY
});

// Create interviewer avatar
const createInterviewer = async (videoUrl: string) => {
  return await tavusClient.replicas.create_new_replica({
    train_video_url: videoUrl,
    replica_name: "AI_Interviewer_v1",
    callback_url: `${process.env.BASE_URL}/api/tavus/callback`
  });
};
```

Integration Timeline: Week 2
Fallback Plan: Static avatar images with voice-only interaction

3.2 ElevenLabs Integration (Voice Synthesis)

API Endpoints:
- `GET /v1/voices` - List available voices

- `POST /v1/text-to-speech/{voice_id}` - Generate speech
- `POST /v1/text-to-speech/stream` - Real-time streaming

Implementation Strategy:

```
// ElevenLabs SDK Integration
import { ElevenLabsAPI } from 'elevenlabs';

const elevenLabs = new ElevenLabsAPI({
  apiKey: process.env.ELEVENLABS_API_KEY
});

// Generate interview question audio
const generateQuestionAudio = async (text: string) => {
  return await elevenLabs.textToSpeech.convert({
    voice_id: "professional_interviewer_voice",
    text: text,
    model_id: "eleven_multilingual_v2",
    voice_settings: {
      stability: 0.7,
      similarity_boost: 0.8
    }
  });
};
```

Integration Timeline: Week 2
Fallback Plan: Browser Web Speech API for basic TTS

3.3 RevenueCat Integration (Subscriptions)

SDK Setup:

```
// RevenueCat Web SDK
import Purchases from '@revenuecat/purchases-js';

Purchases.configure({
  apiKey: process.env.REVENUECAT_PUBLIC_KEY,
  appUserID: user.id
});

// Subscription management
const handleSubscription = async (packageId: string) => {
  try {
    const purchaseResult = await Purchases.purchasePackage(packageId);
    // Handle successful purchase
  } catch (error) {
    // Handle purchase error
  }
};
```

Integration Timeline: Week 3
Fallback Plan: Simple Stripe integration for payments

---

4. Week-by-Week Development Milestones

Week 1: Foundation & Core Setup

Days 1-2: Project Initialization
- [ ] Set up Next.js project with bolt.new
- [ ] Configure TypeScript, Tailwind, and ESLint
- [ ] Set up Supabase project and database schema
- [ ] Implement basic authentication system
- [ ] Create responsive layout and navigation

Days 3-4: Core Interview System
- [ ] Design database schema for questions and interviews
- [ ] Implement question bank management
- [ ] Create basic interview flow UI
- [ ] Add question categorization and filtering
- [ ] Implement interview session state management

Days 5-7: User Management & Profiles
- [ ] Complete user registration and login flows
- [ ] Build user profile management
- [ ] Implement interview history tracking
- [ ] Add basic analytics dashboard
- [ ] Create responsive mobile-first design

Week 1 Deliverables:
- Functional authentication system
- Basic interview question flow
- User profile management
- Mobile-responsive design
- Database schema implemented

Week 2: AI Integration & Enhanced Features

Days 8-9: OpenAI Integration
- [ ] Integrate GPT-4 for dynamic question generation
- [ ] Implement response analysis and scoring
- [ ] Add personalized feedback system
- [ ] Create interview difficulty adjustment

Days 10-11: Tavus Avatar Integration
- [ ] Set up Tavus API integration
- [ ] Create AI interviewer replica
- [ ] Implement video generation workflow
- [ ] Add avatar display in interview interface

Days 12-14: ElevenLabs Voice Integration
- [ ] Integrate ElevenLabs TTS API
- [ ] Implement voice question delivery
- [ ] Add voice response recording
- [ ] Create audio playback controls

Week 2 Deliverables:
- AI-powered question generation
- Working AI avatar integration
- Voice synthesis for questions
- Enhanced interview experience

Week 3: Monetization & Advanced Features

Days 15-16: RevenueCat Integration
- [ ] Set up RevenueCat subscription system
- [ ] Implement free vs premium tiers
- [ ] Add payment processing
- [ ] Create subscription management UI

Days 17-18: Analytics & Reporting
- [ ] Build comprehensive analytics dashboard
- [ ] Implement performance tracking
- [ ] Add progress visualization
- [ ] Create improvement recommendations

Days 19-21: Polish & Optimization
- [ ] Optimize performance and loading times
- [ ] Implement error handling and fallbacks
- [ ] Add comprehensive testing
- [ ] Refine UI/UX based on testing

Week 3 Deliverables:
- Functional subscription system
- Advanced analytics dashboard
- Performance optimizations
- Comprehensive error handling

Week 4: Demo Preparation & Launch

Days 22-23: Demo Content Creation
- [ ] Create compelling demo scenarios
- [ ] Prepare sample interview sessions
- [ ] Record demo videos and screenshots
- [ ] Write demo script and talking points

Days 24-25: Final Testing & Bug Fixes
- [ ] Conduct end-to-end testing
- [ ] Fix critical bugs and issues
- [ ] Optimize for demo environment
- [ ] Prepare backup plans for live demo

Days 26-28: Launch Preparation
- [ ] Deploy to production environment
- [ ] Set up monitoring and analytics
- [ ] Prepare marketing materials
- [ ] Final demo rehearsals

Week 4 Deliverables:
- Production-ready application

- Compelling demo presentation
- Marketing materials
- Launch strategy execution

---

5. Demo Preparation Strategy

Demo Structure (10-minute presentation)

1. **Problem Statement** (1 minute)
   - Current interview preparation challenges
   - Market opportunity and user pain points

2. **Solution Overview** (2 minutes)
   - AI-powered interview coaching platform
   - Key differentiators and value proposition

3. **Live Demo** (5 minutes)
   - User registration and onboarding
   - AI avatar interview simulation
   - Voice interaction demonstration
   - Analytics and progress tracking

4. **Technology Showcase** (1 minute)
   - Sponsor API integrations (Tavus, ElevenLabs, RevenueCat)
   - Technical architecture highlights

5. **Business Model & Future** (1 minute)
   - Subscription tiers and pricing
   - Growth strategy and market expansion

Demo Environment Setup

- **Staging Environment:** Identical to production
- **Demo Data:** Pre-populated with realistic scenarios
- **Backup Plans:** Screenshots and recorded videos
- **Internet Backup:** Mobile hotspot for connectivity issues

Presentation Materials

- **Slide Deck:** 10 slides maximum, visual-heavy
- **Demo Script:** Detailed walkthrough with timing
- **Video Backup:** 3-minute demo video as fallback
- **One-Pager:** Technical and business summary

---

6. Risk Mitigation & Contingency Plans

Technical Risks

Risk 1: API Integration Failures
- **Probability:** Medium

- **Impact:** High
- **Mitigation:**
- Implement fallback systems for each API
- Create mock services for demo purposes
- Test integrations early and frequently

Risk 2: Performance Issues
- **Probability:** Medium
- **Impact:** Medium
- **Mitigation:**
- Implement caching strategies
- Optimize bundle sizes
- Use CDN for static assets

Risk 3: Mobile Compatibility
- **Probability:** Low
- **Impact:** High
- **Mitigation:**
- Test on multiple devices throughout development
- Use responsive design principles
- Implement progressive web app features

Business Risks

Risk 1: Feature Scope Creep
- **Probability:** High
- **Impact:** High
- **Mitigation:**
- Strict adherence to MVP scope
- Weekly feature review meetings
- Clear priority framework

Risk 2: Demo Technical Failures
- **Probability:** Medium
- **Impact:** High
- **Mitigation:**
- Multiple backup plans
- Recorded demo videos
- Offline demo capabilities

Contingency Plans

Plan A: Full Feature Demo
- All integrations working
- Live AI avatar and voice interaction
- Real-time subscription processing

Plan B: Partial Integration Demo
- Core interview functionality
- Static avatar with voice synthesis
- Simulated subscription flow

Plan C: Prototype Demo
- Basic interview flow
- Pre-recorded audio/video
- Mockup subscription interface

---

7. Post-Hackathon SaaS Scaling Roadmap

Phase 1: Market Validation (Months 1-2)

- **User Acquisition:** 1,000 beta users
- **Feature Refinement:** Based on user feedback
- **Performance Optimization:** Sub-2s load times
- **Mobile App Development:** React Native implementation

Phase 2: Product-Market Fit (Months 3-6)

- **Advanced AI Features:** GPT-4 fine-tuning for interviews
- **Industry Specialization:** Tech, finance, healthcare verticals
- **Enterprise Features:** Team management and analytics
- **API Development:** Third-party integrations

Phase 3: Scale & Growth (Months 6-12)

- **Multi-language Support:** Global market expansion
- **Advanced Analytics:** ML-powered insights
- **White-label Solutions:** B2B2C opportunities
- **Acquisition Strategy:** Competitor analysis and positioning

Technical Architecture Evolution

Current (Hackathon):

```
Next.js → Supabase → External APIs
```

Phase 1 (Validation):

```
Next.js + Mobile App → Supabase + Redis → Microservices
```

Phase 2 (Scale):

```
Multi-platform → Distributed Database → AI Pipeline → CDN
```

Revenue Projections

- **Month 1:** $500 MRR (50 premium users)
- **Month 6:** $15,000 MRR (1,500 premium users)
- **Month 12:** $100,000 MRR (10,000 premium users)

---

8. Code Architecture & File Structure

Project Structure

```
mobile-ai-interview-coach/
├── src/
│   ├── app/                    # Next.js App Router
│   │   ├── (auth)/             # Authentication routes
│   │   ├── dashboard/          # User dashboard
│   │   ├── interview/          # Interview interface
│   │   ├── api/                # API routes
│   │   └── globals.css         # Global styles
│   ├── components/             # Reusable components
│   │   ├── ui/                 # Shadcn/ui components
│   │   ├── interview/          # Interview-specific components
│   │   ├── dashboard/          # Dashboard components
│   │   └── common/             # Shared components
│   ├── lib/                    # Utility functions
│   │   ├── api/                # API clients
│   │   ├── auth/               # Authentication utilities
│   │   ├── database/           # Database utilities
│   │   └── utils.ts            # General utilities
│   ├── hooks/                  # Custom React hooks
│   ├── stores/                 # Zustand stores
│   ├── types/                  # TypeScript type definitions
│   └── constants/              # Application constants
├── public/                     # Static assets
├── prisma/                     # Database schema and migrations
├── docs/                       # Documentation
└── tests/                      # Test files
```

Key Component Architecture

Interview System:

```
// stores/interview.ts
interface InterviewState {
  currentQuestion: Question | null;
  responses: Response[];
  score: number;
  isRecording: boolean;
  avatarVideo: string | null;
}

// components/interview/InterviewInterface.tsx
export function InterviewInterface() {
  const { currentQuestion, startInterview } = useInterviewStore();
  const { generateAvatar } = useTavus();
  const { synthesizeVoice } = useElevenLabs();

  return (
    <div className="interview-container">
      <AvatarDisplay />
      <QuestionDisplay />
      <ResponseRecorder />
      <ProgressIndicator />
    </div>
  );
}
```

API Integration Layer:

```
// lib/api/tavus.ts
export class TavusClient {
  private client: Tavus;

  constructor() {
    this.client = new Tavus({
      apiKey: process.env.TAVUS_API_KEY!
    });
  }

  async generateInterviewerResponse(script: string): Promise<string> {
    const video = await this.client.videos.create_video_from_replica_and_script({
      replica_id: process.env.INTERVIEWER_REPLICA_ID!,
      script,
      video_name: `interview_${Date.now()}`
    });

    return video.download_url;
  }
}
```

Database Schema (Prisma)

```
// prisma/schema.prisma
model User {
  id           String   @id @default(cuid())
  email        String   @unique
  name         String?
  createdAt    DateTime @default(now())
  updatedAt    DateTime @updatedAt

  interviews   Interview[]
  subscription Subscription?
}

model Interview {
  id           String   @id @default(cuid())
  userId       String
  user         User     @relation(fields: [userId], references: [id])

  questions    InterviewQuestion[]
  responses    InterviewResponse[]

  score        Float?
  duration     Int?
  completedAt DateTime?
  createdAt    DateTime @default(now())
}

model Question {
  id           String   @id @default(cuid())
  text         String
  category     String
  difficulty   String
  industry     String?

  interviewQuestions InterviewQuestion[]
}

model Subscription {
  id           String   @id @default(cuid())
  userId       String   @unique
  user         User     @relation(fields: [userId], references: [id])

  tier         String   // 'free' | 'premium'
  status       String   // 'active' | 'canceled' | 'expired'

  createdAt    DateTime @default(now())
  updatedAt    DateTime @updatedAt
}
```

Environment Configuration

```
# .env.local
# Database
DATABASE_URL="postgresql://..."
DIRECT_URL="postgresql://..."

# Authentication
SUPABASE_URL="https://..."
SUPABASE_ANON_KEY="..."

# AI Services
OPENAI_API_KEY="sk-..."
TAVUS_API_KEY="..."
ELEVENLABS_API_KEY="..."

# Payments
REVENUECAT_PUBLIC_KEY="..."
STRIPE_PUBLIC_KEY="pk_..."

# App Configuration
NEXT_PUBLIC_APP_URL="http://localhost:3000"
```

Implementation Checklist

Pre-Development Setup

- [ ] Set up development environment
- [ ] Create accounts for all required services
- [ ] Obtain API keys and configure environment variables
- [ ] Set up project repository and version control

Week 1 Checklist

- [ ] Next.js project initialization with bolt.new
- [ ] Supabase project setup and database schema
- [ ] Basic authentication implementation
- [ ] Core UI components and layout
- [ ] Question bank and interview flow

Week 2 Checklist

- [ ] OpenAI integration for AI-powered features
- [ ] Tavus API integration for AI avatars
- [ ] ElevenLabs integration for voice synthesis
- [ ] Enhanced interview experience

Week 3 Checklist

- [ ] RevenueCat subscription system
- [ ] Analytics dashboard implementation
- [ ] Performance optimization
- [ ] Comprehensive testing

Week 4 Checklist

- [ ] Demo preparation and content creation
- [ ] Final testing and bug fixes
- [ ] Production deployment
- [ ] Launch strategy execution

Success Metrics

Hackathon Success Criteria

- **Functional MVP:** Core interview functionality working
- **API Integrations:** At least 2/3 sponsor APIs integrated
- **Demo Quality:** Smooth 10-minute presentation
- **Technical Innovation:** Creative use of AI technologies
- **Business Viability:** Clear monetization strategy

Post-Hackathon KPIs

- **User Acquisition:** 1,000 users in first month
- **Engagement:** 70% completion rate for interviews
- **Conversion:** 10% free-to-premium conversion rate
- **Revenue:** $5,000 MRR by month 3
- **Technical:** 99.9% uptime, <2s load times

Conclusion

This roadmap provides a comprehensive strategy for building a competitive Mobile AI Interview Coach app within the 4-week hackathon timeline. By focusing on core MVP features, leveraging sponsor APIs effectively, and maintaining a clear development schedule, the project is positioned for both hackathon success and future SaaS growth.

The key to success will be disciplined execution, early testing of integrations, and maintaining focus on the core value proposition: helping users improve their interview skills through AI-powered coaching.

Next Steps:
1. Review and approve this roadmap
2. Set up development environment
3. Begin Week 1 implementation
4. Schedule weekly progress reviews
5. Prepare for an exciting hackathon journey!

Last Updated: June 3, 2025
Document Version: 1.0