

Pull Request: 15-Subdomain Guru Network Implementation

Branch: `feat/guru-subdomain-network`

Epic: Monetization & Content Network

Priority: High

Estimated Hours: 40-60 hours

Objective

Implement a complete 15-subdomain network for `yoohoo.guru` with individual Guru characters, content management, affiliate monetization, and seamless integration with the main skill-sharing platform.

Requirements Overview

Core Features to Implement

- 1. Dynamic Subdomain Routing** - Detect and route 15 subdomains to customized experiences
- 2. Guru Character System** - Individual branding per subdomain (Chef Guru, Computer Guru, etc.)
- 3. Content Management System** - CRUD operations for subdomain-specific content
- 4. Affiliate Integration** - Amazon Associates + multiple affiliate networks
- 5. SEO Optimization** - Individual sitemaps, meta tags, structured data per subdomain
- 6. Lead Generation** - Contact forms and professional referral system
- 7. Cross-Platform Integration** - Link subdomains to main coaching platform
- 8. Analytics Tracking** - Individual subdomain performance monitoring

Technical Implementation

1. Subdomain Configuration

File: `backend/src/config/subdomains.js`

```
javascript
```

```
const subdomainConfig = {
  'cooking': {
    character: 'Chef Guru',
    category: 'culinary',
    primarySkills: ['cooking', 'baking', 'nutrition', 'meal-prep'],
    affiliateCategories: ['kitchen-tools', 'cookbooks', 'ingredients'],
    theme: {
      primaryColor: '#e74c3c',
      secondaryColor: '#f39c12',
      icon: '👨‍🍳'
    }
  },
  'music': {
    character: 'Music Guru',
    category: 'audio',
    primarySkills: ['guitar', 'piano', 'vocals', 'production'],
    affiliateCategories: ['instruments', 'equipment', 'software'],
    theme: {
      primaryColor: '#9b59b6',
      secondaryColor: '#8e44ad',
      icon: '🎵'
    }
  },
  // ... (continue for all 15 subdomains)
};

export default subdomainConfig;
```

2. Middleware Implementation

File: [backend/src/middleware/subdomainHandler.js](#)

javascript

```
import subdomainConfig from './config/subdomains.js';

export const subdomainHandler = (req, res, next) => {
  // Extract subdomain from request
  const subdomain = req.get('host').split('.')[0];

  if (subdomain && subdomainConfig[subdomain]) {
    req.guru = {
      subdomain,
      config: subdomainConfig[subdomain],
      isGuru: true
    };
  } else {
    req.guru = {
      isGuru: false,
      isMainSite: true
    };
  }

  next();
};
```

3. Database Schema Updates

Firebase Realtime Database Structure:

json

```
{  
  "gurus": {  
    "cooking": {  
      "character": "Chef Guru",  
      "posts": {  
        "post_001": {  
          "title": "10 Essential Knife Skills Every Cook Needs",  
          "slug": "essential-knife-skills-cooking",  
          "content": "...",  
          "excerpt": "Master these fundamental knife techniques...",  
          "publishedAt": 1672531200000,  
          "author": "Chef Guru",  
          "tags": ["knives", "basics", "cooking"],  
          "seo": {  
            "metaTitle": "10 Essential Knife Skills - Chef Guru",  
            "metaDescription": "Master essential knife skills...",  
            "canonicalUrl": "https://cooking.yoohoo.guru/essential-knife-skills-cooking"  
          },  
          "affiliates": {  
            "primary": "https://amzn.to/chef-knife-set",  
            "secondary": "https://williams-sonoma.com/...",  
            "courseLink": "https://masterclass.com/gordon-ramsay"  
          },  
          "leadForms": {  
            "cookingLessons": true,  
            "personalChef": true  
          },  
          "analytics": {  
            "views": 2847,  
            "affiliateClicks": 127,  
            "leads": 8  
          }  
        }  
      }  
    },  
    "pages": {  
      "about": {  
        "title": "About Chef Guru",  
        "content": "Meet Chef Guru, your culinary companion...",  
        "cta": "Book a Cooking Session"  
      },  
      "services": {  
        "title": "Cooking Services",  
        "services": [  

```

```

    {
      "name": "Personal Cooking Lessons",
      "price": 75,
      "description": "One-on-one cooking instruction"
    }
  ]
}
},
"leads": {
  "lead_001": {
    "name": "John Smith",
    "email": "john@example.com",
    "service": "cooking-lessons",
    "message": "Interested in Italian cooking classes",
    "createdAt": 1672531200000,
    "status": "new",
    "value": 200
  }
}
}
}

// Repeat structure for all 15 subdomains
}
}
}

```

4. Frontend Routing Updates

File: `frontend/src/components/AppRouter.js`

Add these routes:

```

jsx

// Dynamic subdomain routing
<Route path="/" element={req.guru?.isGuru ? <GuruHomePage /> : <MainHomePage />} />
<Route path="/about" element={req.guru?.isGuru ? <GuruAboutPage /> : <MainAboutPage />} />
<Route path="/blog" element={req.guru?.isGuru ? <GuruBlogPage /> : <Redirect to="/skills" />} />
<Route path="/blog/:slug" element={<GuruBlogPostPage />} />
<Route path="/services" element={<GuruServicesPage />} />
<Route path="/contact" element={<GuruContactPage />} />
<Route path="/book-session" element={<BookingPage />} />

```

5. New React Components

GuruHomePage Component

File: frontend/src/screens/GuruHomePage.js

```
jsx

import React, { useEffect, useState } from 'react';
import { useGuru } from './hooks/useGuru';
import HeroSection from './components/guru/HeroSection';
import FeaturedPosts from './components/guru/FeaturedPosts';
import ServicesSection from './components/guru/ServicesSection';
import NewsletterSignup from './components/guru/NewsletterSignup';

export default function GuruHomePage() {
  const { guru, posts, loading } = useGuru();

  if (loading) return <LoadingSpinner />

  return (
    <div className="guru-homepage">
      <HeroSection
        character={guru.character}
        theme={guru.theme}
        primarySkills={guru.primarySkills}
      />
      <FeaturedPosts posts={posts.slice(0, 6)} />
      <ServicesSection category={guru.category} />
      <NewsletterSignup guruName={guru.character} />
    </div>
  );
}

}
```

GuruBlogPage Component

File: frontend/src/screens/GuruBlogPage.js

```
jsx
```

```
import React, { useState, useEffect } from 'react';
import { useGuru } from './hooks/useGuru';
import BlogGrid from './components/guru/BlogGrid';
import CategoryFilter from './components/guru/CategoryFilter';
import SearchBar from './components/guru/SearchBar';

export default function GuruBlogPage() {
  const { guru, posts, loading } = useGuru();
  const [filteredPosts, setFilteredPosts] = useState([]);
  const [searchTerm, setSearchTerm] = useState("");
  const [selectedTag, setSelectedTag] = useState('all');

  // Filtering and search logic
  useEffect(() => {
    let filtered = posts;

    if (searchTerm) {
      filtered = filtered.filter(post =>
        post.title.toLowerCase().includes(searchTerm.toLowerCase()) ||
        post.content.toLowerCase().includes(searchTerm.toLowerCase())
      );
    }

    if (selectedTag !== 'all') {
      filtered = filtered.filter(post =>
        post.tags?.includes(selectedTag)
      );
    }

    setFilteredPosts(filtered);
  }, [posts, searchTerm, selectedTag]);

  return (
    <div className="guru-blog-page">
      <div className="blog-header">
        <h1>{guru.character} Knowledge Hub</h1>
        <SearchBar value={searchTerm} onChange={setSearchTerm} />
      </div>

      <div className="blog-content">
        <aside className="blog-sidebar">
          <CategoryFilter
            selectedTag={selectedTag}
          >
        </CategoryFilter>
      </div>
    </div>
  );
}
```

```
onTagSelect={setSelectedTag}
availableTags={guru.primarySkills}
/>
</aside>

<main className="blog-main">
  <BlogGrid posts={filteredPosts} />
</main>
</div>
</div>
);
}
```

6. API Endpoints

Guru Content API

File: [backend/src/routes/gurus.js](#)

javascript

```
import express from 'express';
import { db } from './config.firebaseio.js';
import { subdomainHandler } from './middleware/subdomainHandler.js';

const router = express.Router();

// Get guru homepage data
router.get('/:subdomain/home', subdomainHandler, async (req, res) => {
  try {
    const { subdomain } = req.params;

    const guruData = await db.ref(`gurus/${subdomain}`).once('value');
    const posts = await db.ref(`gurus/${subdomain}/posts`)
      .orderByChild('publishedAt')
      .limitToFirst(6)
      .once('value');

    res.json({
      guru: guruData.val(),
      featuredPosts: Object.values(posts.val() || {}),
      success: true
    });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

// Get all blog posts for subdomain
router.get('/:subdomain/posts', async (req, res) => {
  try {
    const { subdomain } = req.params;
    const { page = 1, limit = 12, tag, search } = req.query;

    let query = db.ref(`gurus/${subdomain}/posts`)
      .orderByChild('publishedAt');

    const posts = await query.once('value');
    let postsArray = Object.values(posts.val() || {});

    const posts = await query.once('value');
    let postsArray = Object.values(posts.val() || {});

    // Apply filters
    if (tag && tag !== 'all') {
      postsArray = postsArray.filter(post =>
        post.tags?.includes(tag)
      );
    }

    res.json({
      posts: postsArray,
      success: true
    });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});
```

```
);

}

if (search) {
  postsArray = postsArray.filter(post =>
    post.title.toLowerCase().includes(search.toLowerCase()) ||
    post.excerpt.toLowerCase().includes(search.toLowerCase())
  );
}

// Pagination
const startIndex = (page - 1) * limit;
const endIndex = startIndex + parseInt(limit);
const paginatedPosts = postsArray.slice(startIndex, endIndex);

res.json({
  posts: paginatedPosts,
  totalPosts: postsArray.length,
  totalPages: Math.ceil(postsArray.length / limit),
  currentPage: parseInt(page),
  success: true
});

} catch (error) {
  res.status(500).json({ error: error.message });
}
});

// Submit lead form
router.post('/:subdomain/leads', async (req, res) => {
  try {
    const { subdomain } = req.params;
    const { name, email, service, message } = req.body;

    const leadId = db.ref().push().key;
    const leadData = {
      id: leadId,
      name,
      email,
      service,
      message,
      subdomain,
      createdAt: Date.now(),
      status: 'new'
    };
  }
});
```

```
await db.ref(`gurus/${subdomain}/leads/${leadId}`).set(leadData);

// Send notification email
await sendLeadNotification(subdomain, leadData);

res.json({ success: true, message: 'Lead submitted successfully' });
} catch (error) {
  res.status(500).json({ error: error.message });
}
});

export default router;
```

7. Custom Hooks

File: [frontend/src/hooks/useGuru.js](#)

javascript

```
import { useState, useEffect } from 'react';
import { useParams, useLocation } from 'react-router-dom';
import { getGuruData, getGuruPosts } from './services/guruService';

export function useGuru() {
  const [guru, setGuru] = useState(null);
  const [posts, setPosts] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  const location = useLocation();
  const subdomain = window.location.hostname.split('.')[0];

  useEffect(() => {
    const fetchGuruData = async () => {
      try {
        setLoading(true);

        const [guruData, postsData] = await Promise.all([
          getGuruData(subdomain),
          getGuruPosts(subdomain)
        ]);

        setGuru(guruData.guru);
        setPosts(postsData.posts || []);
      } catch (err) {
        setError(err.message);
      } finally {
        setLoading(false);
      }
    };

    if (subdomain && subdomain !== 'www' && subdomain !== 'yoohoo') {
      fetchGuruData();
    }
  }, [subdomain, location.pathname]);

  return {
    guru,
    posts,
    loading,
    error,
    subdomain,
  };
}
```

```
    isGuruSite: !['www', 'yoohoo'].includes(subdomain)
  );
}
```

SEO Implementation

Sitemap Generation

File: [backend/src/utils/sitemapGenerator.js](#)

javascript

```
export async function generateSubdomainSitemap(subdomain) {
  const posts = await getGuruPosts(subdomain);
  const baseUrl = `https://${subdomain}.yoohoo.guru`;

  const urls = [
    { url: baseUrl, priority: 1.0 },
    { url: `${baseUrl}/about`, priority: 0.8 },
    { url: `${baseUrl}/services`, priority: 0.9 },
    { url: `${baseUrl}/blog`, priority: 0.8 },
    { url: `${baseUrl}/contact`, priority: 0.7 },
    ...posts.map(post => ({
      url: `${baseUrl}/blog/${post.slug}`,
      lastmod: new Date(post.publishedAt).toISOString(),
      priority: 0.6
    }))
  ];
}

return generateXMLSitemap(urls);
}
```

Affiliate Integration

Amazon Associates Integration

File: [backend/src/services/affiliateService.js](#)

javascript

```
export class AffiliateService {
  static generateAmazonLink(asin, tag = 'yoohooru-20') {
    return `https://www.amazon.com/dp/${asin}?tag=${tag}`;
  }

  static async trackAffiliateClick(subdomain, linkType, productId) {
    const clickData = {
      subdomain,
      linkType,
      productId,
      timestamp: Date.now(),
      ip: req.ip,
      userAgent: req.get("User-Agent")
    };

    await db.ref(`analytics/affiliate-clicks`).push(clickData);
  }
}
```

Make.com Integration Setup

Webhook Endpoints for Make

File: [backend/src/routes/webhooks/make.js](#)

javascript

```

// Content generation webhook
router.post('/generate-content/:subdomain', async (req, res) => {
  const { subdomain } = req.params;
  const { topic, keywords, targetLength } = req.body;

  // Generate content using OpenRouter
  const content = await generateGuruContent(subdomain, topic, keywords, targetLength);

  // Auto-publish to Firebase
  await publishGuruPost(subdomain, content);

  res.json({ success: true, postId: content.id });
});

// Affiliate performance webhook
router.get('/affiliate-report/:subdomain', async (req, res) => {
  const { subdomain } = req.params;
  const report = await generateAffiliateReport(subdomain);
  res.json(report);
});

```

Testing Strategy

Unit Tests

- Subdomain routing middleware
- Guru content API endpoints
- Affiliate link generation
- Lead form processing

Integration Tests

- Complete guru homepage load
- Blog post rendering with affiliates
- Lead submission flow
- Cross-linking to main platform

E2E Tests

- User journey: subdomain → blog → affiliate click → main platform booking
- SEO meta tag generation

- Mobile responsiveness

Analytics & Monitoring

Metrics to Track

javascript

```
// Per subdomain metrics
const metrics = {
  traffic: {
    pageViews: 0,
    uniqueVisitors: 0,
    sessionDuration: 0,
    bounceRate: 0
  },
  conversion: {
    affiliateClicks: 0,
    affiliateConversions: 0,
    leads: 0,
    bookings: 0
  },
  revenue: {
    affiliateEarnings: 0,
    directBookings: 0,
    leadValue: 0
  }
};
```

Deployment Checklist

Pre-Deployment

- DNS wildcard configured in Porkbun
- SSL certificate covers all subdomains
- Firebase security rules updated for guru data
- Environment variables set for affiliate IDs
- Make.com scenarios configured and tested

Post-Deployment

- Verify all 15 subdomains resolve correctly
- Test content generation pipeline
- Verify affiliate tracking working

- Submit sitemaps to Google Search Console
- Monitor error logs for subdomain issues

Content Strategy

Launch Content (5 posts per subdomain = 75 total)

Each subdomain should launch with:

1. "Ultimate Guide to [Skill]" (pillar content)
2. "10 Essential Tools for [Skill]" (affiliate-heavy)
3. "[Skill] for Beginners: Complete Roadmap"
4. "Common [Skill] Mistakes to Avoid"
5. "How to Find the Right [Skill] Teacher" (CTA to main platform)

Content Calendar

- Week 1: Publish all pillar content (15 posts)
- Week 2: Publish tool/equipment guides (15 posts)
- Week 3: Publish beginner guides (15 posts)
- Week 4: Publish mistake articles (15 posts)
- Week 5: Publish teacher-finding guides (15 posts)

Success Metrics

30-Day Goals

- All 15 subdomains live and functional
- 75 high-quality blog posts published
- 500+ organic search clicks across network
- 50+ affiliate clicks generated
- 25+ leads captured
- 5+ bookings from subdomain traffic

90-Day Goals

- 10,000+ monthly organic visitors across network
- \$500+ monthly affiliate revenue
- 200+ monthly leads captured
- 50+ monthly bookings from subdomains

Revenue Projections

Conservative Monthly Targets (Month 6)

Affiliate Revenue: \$2,500/month

Lead Generation: \$1,500/month

Direct Bookings: \$3,000/month

Total per Subdomain: \$467/month

Network Total: \$7,000/month

This comprehensive implementation creates a powerful content network that drives traffic and revenue while seamlessly integrating with the main yoohoo.guru coaching platform. The Make.com integration automates content creation and performance tracking, while the guru-specific branding creates authority in each niche.