lib/

├── auth/

│ ├── models/

│ │ ├── app\_user.dart

│ │ ├── authentication\_models.dart

│ │ ├── followers\_sub.dart

│ │ ├── following\_sub.dart

│ │ ├── resume\_sub.dart

│ │ └── saved\_post.dart

│ └── auth\_gate.dart

├── chat/

│ ├── models/

│ │ ├── conversation.dart

│ │ └── message.dart

│ ├── chat\_screen.dart

│ ├── messages\_screen.dart

│ └── user\_search\_screen.dart

├── config/

│ ├── .gitignore

│ ├── config.dart

│ ├── decorations.dart

│ ├── firebase\_options.dart

│ └── routes.dart

├── constants/

│ └── notification\_types.dart

├── core/

│ ├── handlers/

│ │ ├── Company\_profile/

│ │ │ ├── company\_upload\_image.dart

│ │ │ ├── company\_upload\_image\_html.dart

│ │ │ └── company\_upload\_image\_io.dart

│ │ ├── image\_uploader/

│ │ │ ├── RecipeImageUploader.dart

│ │ │ ├── RecipeImageUploaderHtml.dart

│ │ │ └── RecipeImageUploaderIo.dart

│ │ ├── maps/

│ │ ├── postimage\_uploader/

│ │ │ ├── upload\_image.dart

│ │ │ ├── upload\_image\_html.dart

│ │ │ └── upload\_image\_io.dart

│ │ └── image\_handler.dart

│ ├── recipe\_provider.dart

│ └── storage\_service.dart

├── create/

│ ├── job\_create.dart

│ ├── post\_create.dart

│ └── recipe\_create.dart

├── deep\_linking/

│ ├── branch\_link\_generator.dart

│ ├── branch\_observer.dart

│ ├── branch\_service.dart

│ └── link\_generator.dart

├── home/

│ ├── app\_bar\_actions.dart

│ ├── create\_drawer.dart

│ ├── home\_screen.dart

│ ├── large\_screen\_layout.dart

│ └── small\_screen\_layout.dart

├── models/

│ ├── jobs\_model/

│ │ ├── applicants\_sub.dart

│ │ ├── job.dart

│ │ └── Subcollection.txt

│ ├── posts/

│ │ └── posts.dart

│ ├── recipes/

│ │ └── recipe.dart

│ └── notification\_model.dart

├── screens/

│ ├── single/

│ │ └── recipe\_detail\_screen.dart

│ ├── job\_feed\_screen.dart

│ ├── notifications\_screen.dart

│ ├── post\_feed\_screen.dart

│ └── recipe\_feed\_screen.dart

├── services/

│ ├── FCMService.dart

│ └── notification\_service.dart

├── theme/

│ └── theme.dart

├── user/

│ ├── EditUserProfile\_Screen.dart

│ ├── ResumeBuilderScreen.dart

│ ├── ResumeViewScreen.dart

│ ├── UserProfile\_Screen.dart

│ └── users.dart

├── utils/

│ └── navigation\_helper.dart

├── widgets/

│ ├── ads\_widget.dart

│ ├── blogs\_tab.dart

│ ├── friend\_suggestions\_widget.dart

│ ├── jobs\_tab.dart

│ ├── notification\_icon.dart

│ ├── notification\_tile.dart

│ ├── post\_card.dart

│ ├── posts\_tab.dart

│ ├── recipes\_tab.dart

│ └── user\_info\_panel.dart

├── app.dart

├── deep\_linking.dart

├── firebase\_options.dart

├── imports.dart

└── main.dart

Here's a comprehensive project plan for CuliNect:

1. Core Platform Features

* Authentication & User Management
* Social Feed & Interactions
* Messaging & Notifications
* Profile Management
* Search & Discovery
* Mobile & Web Responsiveness

1. Professional Profiles

* Resume Builder System
  + Multiple templates
  + PDF export/sharing
  + ATS optimization
  + Portfolio integration
  + Application tracking
* Professional Verification
* Skills & Certifications
* Work History
* Portfolio Gallery

1. Company Management

* Company Profiles
* Verification System
* Branch Management
* Job Posting System
* Analytics Dashboard
* Company Reviews
* Industry Insights

1. Recipe Management

* Recipe Creation & Sharing
* Nutrition Calculator
  + Automatic calculations
  + Serving adjustments
  + Visual nutrition labels
  + Dietary tracking
* Version Control
* Collaboration Tools
* Print & Share Options

1. CuliKnowledge Hub

* Ingredient Encyclopedia
* Dish Database
* Technique Library
* Equipment Guide
* Cultural Context
* Sustainability Info
* Wine Knowledge
* Food Science

1. CuliQuiz System

* Multiple Categories
* Difficulty Levels
* Progress Tracking
* Certifications
* Leaderboards
* Achievement System
* Visual Assessments
* Timed Challenges

1. CuliAsk Forum

* Q&A Platform
* Expert Verification
* Topic Categories
* Solution Marking
* Search Optimization
* Community Moderation
* Rich Media Support

1. Professional Tools

* Inventory Management
* Cost Calculator
* Menu Planning
* Team Management
* HACCP Compliance
* Equipment Maintenance
* Supplier Directory

1. Community Features

* Challenges & Competitions
* Virtual Events
* Mentorship Program
* Collaborative Cooking
* Industry Updates
* Trending Topics
* Regional Communities

1. Business Features

* Job Board
* Marketplace
* Advertising System
* Premium Memberships
* Business Analytics
* Promotion Tools
* Verification Services

Development Phases:

Phase 1 (Months 1-3):

* Core Platform Development
* Basic Profile System
* Recipe Management
* Essential Social Features

Phase 2 (Months 4-6):

* Professional Profiles
* Company Management
* Job System
* Nutrition Calculator

Phase 3 (Months 7-9):

* Knowledge Hub
* Quiz System
* Forum Implementation
* Professional Tools

Phase 4 (Months 10-12):

* Community Features
* Business Features
* Advanced Analytics
* Platform Optimization

Technical Requirements:

* Flutter/Dart
* Firebase Backend
* Cloud Functions
* Real-time Database
* Secure Authentication
* CDN Integration
* Analytics Tools
* Payment Processing

Phase 1 (Core Development)

1. Authentication System

- Social login (Google, Facebook, Apple)

- Email verification

- Phone verification

- Password recovery

- Session management

- Device tracking

- Login analytics

1. Profile System

- Basic profile setup

- Profile completion tracking

- Profile verification

- Image upload/cropping

- Social links integration

- Activity history

- Privacy settings

1. Recipe Management

- Recipe creation workflow

- Ingredient database integration

- Step-by-step builder

- Image handling

- Cooking time calculator

- Serving size adjuster

- Tags and categories

- Search indexing

1. Social Features

- Feed algorithm

- Post creation (text, image, video)

- Comments system

- Reaction system

- Share functionality

- Save/bookmark system

- Follow system

- Blocking/reporting

Phase 2 (Professional Features)

1. Professional Profiles

- Resume builder

- Templates

- PDF generation

- Online sharing

- Application tracking

- Portfolio management

- Skill endorsements

- Work verification

- Achievement system

1. Company Management

- Company profiles

- Verification process

- Branch management

- Team hierarchy

- Analytics dashboard

- Review system

- Rating algorithm

- Location management

1. Job System

- Job posting

- Template system

- Requirements builder

- Salary calculator

- Application tracking

- Candidate matching

- Interview scheduling

- Status updates

1. Nutrition Calculator

- Ingredient database

- Portion calculator

- Nutrient tracking

- Label generator

- Dietary restrictions

- Allergen warnings

- Export functionality

Phase 3 (Knowledge & Learning)

1. Knowledge Hub

- Ingredient encyclopedia

- Detailed profiles

- Usage examples

- Storage guidelines

- Seasonal tracking

- Technique library

- Video integration

- Step-by-step guides

- Expert tips

- Equipment guide

- Comparisons

- Reviews

- Maintenance tips

1. Quiz System

- Question bank

- Category management

- Difficulty scaling

- Progress tracking

- Certificate generation

- Performance analytics

- Leaderboards

1. Forum Implementation

- Question categories

- Answer ranking

- Expert verification

- Search optimization

- Content moderation

- Rich text editor

- Media integration

1. Professional Tools

- Inventory tracker

- Cost calculator

- Menu planner

- Team scheduler

- Compliance checker

- Equipment maintenance

- Supplier directory

Phase 4 (Community & Business)

1. Community Features

- Challenge system

- Event management

- Mentorship matching

- Live streaming

- Virtual workshops

- Community polls

- Regional groups

1. Business Features

- Marketplace

- Product listings

- Order management

- Payment processing

- Shipping integration

- Premium features

- Subscription management

- Feature access

- Analytics

1. Advanced Analytics

- User behavior

- Content performance

- Business metrics

- Job market trends

- Industry insights

- Engagement tracking

1. Platform Optimization

- Performance monitoring

- Load balancing

- Cache management

- CDN integration

- Security updates

- API optimization

Technical Infrastructure:

1. Backend Services

- Firebase Authentication

- Cloud Firestore

- Cloud Functions

- Real-time Database

- Cloud Storage

- Analytics

1. Frontend Architecture

- Responsive layouts

- State management

- Offline support

- Image optimization

- Deep linking

- Push notifications

1. Security Implementation

- Data encryption

- Access control

- Content moderation

- Fraud detection

- Backup systems

1. Integration Services

- Payment gateways

- Email services

- SMS services

- Map services

- Social APIs

- Analytics tools

lib/

├── onboarding/

│ ├── onboarding\_flow.dart <-- Create here

│ ├── controller/

│ │ └── onboarding\_controller.dart

│ ├── screens/

│ │ ├── welcome\_screen.dart

│ │ ├── basic\_profile\_screen.dart

│ │ ├── professional\_details\_screen.dart

│ │ └── interests\_screen.dart

│ └── widgets/

│ ├── profile\_image\_picker.dart

│ ├── specialty\_selector.dart

│ └── interest\_grid.dart

**New Project Setup**

**AppUser Model Class**

class AppUser {

final String uid;

final String email;

final String displayName;

final String photoURL;

final String phoneNumber;

final String professionalTitle;

final String currentWorkplace;

final String location;

final List<String> specialties;

final int yearsOfExperience;

final List<String> certifications;

final List<String> education;

final Map<String, String> socialLinks;

final List<String> interests;

final bool isProfileComplete;

final DateTime createdAt;

final DateTime lastUpdated;

AppUser({

required this.uid,

required this.email,

required this.displayName,

this.photoURL = '',

this.phoneNumber = '',

this.professionalTitle = '',

this.currentWorkplace = '',

this.location = '',

this.specialties = const [],

this.yearsOfExperience = 0,

this.certifications = const [],

this.education = const [],

this.socialLinks = const {},

this.interests = const [],

this.isProfileComplete = false,

required this.createdAt,

required this.lastUpdated,

});

factory AppUser.fromMap(Map<String, dynamic> map) {

return AppUser(

uid: map['uid'] ?? '',

email: map['email'] ?? '',

displayName: map['displayName'] ?? '',

photoURL: map['photoURL'] ?? '',

phoneNumber: map['phoneNumber'] ?? '',

professionalTitle: map['professionalTitle'] ?? '',

currentWorkplace: map['currentWorkplace'] ?? '',

location: map['location'] ?? '',

specialties: List<String>.from(map['specialties'] ?? []),

yearsOfExperience: map['yearsOfExperience'] ?? 0,

certifications: List<String>.from(map['certifications'] ?? []),

education: List<String>.from(map['education'] ?? []),

socialLinks: Map<String, String>.from(map['socialLinks'] ?? {}),

interests: List<String>.from(map['interests'] ?? []),

isProfileComplete: map['isProfileComplete'] ?? false,

createdAt: (map['createdAt'] as Timestamp).toDate(),

lastUpdated: (map['lastUpdated'] as Timestamp).toDate(),

);

}

Map<String, dynamic> toMap() {

return {

'uid': uid,

'email': email,

'displayName': displayName,

'photoURL': photoURL,

'phoneNumber': phoneNumber,

'professionalTitle': professionalTitle,

'currentWorkplace': currentWorkplace,

'location': location,

'specialties': specialties,

'yearsOfExperience': yearsOfExperience,

'certifications': certifications,

'education': education,

'socialLinks': socialLinks,

'interests': interests,

'isProfileComplete': isProfileComplete,

'createdAt': Timestamp.fromDate(createdAt),

'lastUpdated': Timestamp.fromDate(lastUpdated),

};

}

AppUser copyWith({

String? displayName,

String? photoURL,

String? phoneNumber,

String? professionalTitle,

String? currentWorkplace,

String? location,

List<String>? specialties,

int? yearsOfExperience,

List<String>? certifications,

List<String>? education,

Map<String, String>? socialLinks,

List<String>? interests,

bool? isProfileComplete,

}) {

return AppUser(

uid: this.uid,

email: this.email,

displayName: displayName ?? this.displayName,

photoURL: photoURL ?? this.photoURL,

phoneNumber: phoneNumber ?? this.phoneNumber,

professionalTitle: professionalTitle ?? this.professionalTitle,

currentWorkplace: currentWorkplace ?? this.currentWorkplace,

location: location ?? this.location,

specialties: specialties ?? this.specialties,

yearsOfExperience: yearsOfExperience ?? this.yearsOfExperience,

certifications: certifications ?? this.certifications,

education: education ?? this.education,

socialLinks: socialLinks ?? this.socialLinks,

interests: interests ?? this.interests,

isProfileComplete: isProfileComplete ?? this.isProfileComplete,

createdAt: this.createdAt,

lastUpdated: DateTime.now(),

);

}

}   
  
Here's where we stand:

1. We've implemented the complete AppUser model with proper nested structure
2. Updated the OnboardingController to handle the new user model
3. Started updating the BasicProfileScreen to work with the nested user properties

Next session we'll:

1. Complete the BasicProfileScreen updates
2. Ensure all screens properly access the nested user properties
3. Test the complete onboarding flow
4. Implement any remaining error handling

Ready to continue implementing these features in our next session!