



Competitive Programming & Learning Platform: Complete Project Structure

Based on your detailed specifications, you have designed a **comprehensive competitive programming and educational platform** that combines the best features of LeetCode, CodeChef, Codeforces, and GeeksforGeeks, with additional focus on college communities, mentorship, and gamified learning experiences. This is essentially a **"RankQuest" or "CodeAcademy Pro" style platform** that serves as a complete coding education ecosystem.

Project Type Classification

- **Primary Type:** Competitive Programming & Code Practice Platform
- **Secondary Features:** Educational Technology (EdTech) Platform
- **Target Audience:** College students, competitive programmers, interview candidates
- **Core Functionality:** Code execution, problem solving, contest hosting, progress tracking

Complete Project Structure

Backend Architecture (Spring Boot)

```
backend/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   ├── com/
│   │   │   │   ├── rankquest/
│   │   │   │   │   ├── RankQuestApplication.java
│   │   │   │   │   ├── config/
│   │   │   │   │   │   ├── SecurityConfig.java
│   │   │   │   │   │   ├── DatabaseConfig.java
│   │   │   │   │   │   ├── RedisConfig.java
│   │   │   │   │   │   ├── JwtConfig.java
│   │   │   │   │   │   ├── WebSocketConfig.java
│   │   │   │   │   │   ├── S3Config.java
│   │   │   │   │   │   ├── DockerConfig.java
│   │   │   │   │   │   └── SwaggerConfig.java
│   │   │   │   │   ├── security/
│   │   │   │   │   │   ├── JwtAuthenticationFilter.java
│   │   │   │   │   │   ├── JwtTokenProvider.java
│   │   │   │   │   │   ├── CustomUserDetailsService.java
│   │   │   │   │   │   ├── OAuth2AuthenticationHandler.java
│   │   │   │   │   │   └── SecurityUtils.java
│   │   │   │   │   └── auth/
```

```
├── controller/
│   └── AuthController.java
├── service/
│   ├── AuthService.java
│   ├── AuthServiceImpl.java
│   ├── JwtService.java
│   └── OAuth2Service.java
├── dto/
│   ├── LoginRequest.java
│   ├── RegisterRequest.java
│   ├── AuthResponse.java
│   ├── RefreshTokenRequest.java
│   └── PasswordResetRequest.java
├── model/
│   ├── User.java
│   ├── RefreshToken.java
│   ├── PasswordResetToken.java
│   └── Role.java
└── repository/
    ├── UserRepository.java
    ├── RefreshTokenRepository.java
    └── PasswordResetTokenRepository.java

├── onboarding/
│   ├── controller/
│   │   └── OnboardingController.java
│   ├── service/
│   │   ├── OnboardingService.java
│   │   └── OnboardingServiceImpl.java
│   ├── dto/
│   │   ├── OnboardingRequest.java
│   │   ├── OnboardingResponse.java
│   │   └── OnboardingUpdateRequest.java
│   ├── model/
│   │   ├── UserProfile.java
│   │   ├── CompetitiveProgrammingProfile.java
│   │   └── AcademicInfo.java
│   └── repository/
│       ├── UserProfileRepository.java
│       └── CompetitiveProgrammingProfileRepository.java

├── profile/
│   ├── controller/
│   │   └── ProfileController.java
│   ├── service/
│   │   ├── ProfileService.java
│   │   ├── ProfileServiceImpl.java
│   │   ├── AvatarService.java
│   │   └── ResumeService.java
│   ├── dto/
│   │   ├── ProfileResponse.java
│   │   ├── ProfileUpdateRequest.java
│   │   ├── ContributionGraphResponse.java
│   │   ├── ActivityResponse.java
│   │   └── StatsResponse.java
│   └── model/
│       ├── UserActivity.java
│       └── ContributionData.java
```

```
└─ Badge.java
└─ problems/
  └─ controller/
    └─ ProblemController.java
    └─ SubmissionController.java
    └─ TestingController.java
  └─ service/
    └─ ProblemService.java
    └─ ProblemServiceImpl.java
    └─ SubmissionService.java
    └─ SubmissionServiceImpl.java
    └─ CodeExecutionService.java
    └─ CodeExecutionServiceImpl.java
    └─ TestCaseService.java
    └─ PlaygroundService.java
  └─ dto/
    └─ ProblemRequest.java
    └─ ProblemResponse.java
    └─ ProblemListResponse.java
    └─ SubmissionRequest.java
    └─ SubmissionResponse.java
    └─ TestRunRequest.java
    └─ TestRunResponse.java
    └─ ExecutionResult.java
    └─ CodeTestRequest.java
  └─ model/
    └─ Problem.java
    └─ ProblemTag.java
    └─ TestCase.java
    └─ Submission.java
    └─ SubmissionResult.java
    └─ Language.java
    └─ Difficulty.java
  └─ repository/
    └─ ProblemRepository.java
    └─ ProblemTagRepository.java
    └─ TestCaseRepository.java
    └─ SubmissionRepository.java
    └─ LanguageRepository.java
└─ sheets/
  └─ controller/
    └─ SheetController.java
  └─ service/
    └─ SheetService.java
    └─ SheetServiceImpl.java
    └─ BookmarkService.java
    └─ ProgressTrackingService.java
  └─ dto/
    └─ SheetRequest.java
    └─ SheetResponse.java
    └─ SheetProgressResponse.java
    └─ BookmarkRequest.java
    └─ CustomListRequest.java
  └─ model/
    └─ Sheet.java
    └─ SheetProblem.java
```

```
├── UserSheetProgress.java
├── Bookmark.java
├── CustomList.java
├── repository/
│   ├── SheetRepository.java
│   ├── SheetProblemRepository.java
│   ├── UserSheetProgressRepository.java
│   ├── BookmarkRepository.java
│   └── CustomListRepository.java
├── contests/
│   ├── controller/
│   │   ├── ContestController.java
│   │   ├── ContestParticipationController.java
│   │   └── ContestAnalyticsController.java
│   ├── service/
│   │   ├── ContestService.java
│   │   ├── ContestServiceImpl.java
│   │   ├── ContestParticipationService.java
│   │   ├── ContestRatingService.java
│   │   ├── ContestSchedulerService.java
│   │   └── LiveContestService.java
│   ├── dto/
│   │   ├── ContestRequest.java
│   │   ├── ContestResponse.java
│   │   ├── ContestListResponse.java
│   │   ├── ParticipationRequest.java
│   │   ├── ParticipationResponse.java
│   │   ├── ContestRankingResponse.java
│   │   ├── ContestAnalyticsResponse.java
│   │   └── LiveContestStatusResponse.java
│   ├── model/
│   │   ├── Contest.java
│   │   ├── ContestProblem.java
│   │   ├── ContestParticipation.java
│   │   ├── ContestSubmission.java
│   │   ├── ContestRanking.java
│   │   ├── UserRating.java
│   │   └── ContestType.java
│   └── repository/
│       ├── ContestRepository.java
│       ├── ContestProblemRepository.java
│       ├── ContestParticipationRepository.java
│       ├── ContestSubmissionRepository.java
│       ├── ContestRankingRepository.java
│       └── UserRatingRepository.java
├── leaderboard/
│   ├── controller/
│   │   └── LeaderboardController.java
│   ├── service/
│   │   ├── LeaderboardService.java
│   │   ├── LeaderboardServiceImpl.java
│   │   ├── GlobalRankingService.java
│   │   ├── CollegeRankingService.java
│   │   └── FriendsRankingService.java
│   ├── dto/
│   │   └── LeaderboardResponse.java
```

```
├── RankingEntry.java
├── GlobalLeaderboardResponse.java
├── CollegeLeaderboardResponse.java
├── FriendsLeaderboardResponse.java
├── model/
│   ├── GlobalRanking.java
│   ├── CollegeRanking.java
│   ├── MonthlyRanking.java
│   └── RankingHistory.java
├── repository/
│   ├── GlobalRankingRepository.java
│   ├── CollegeRankingRepository.java
│   ├── MonthlyRankingRepository.java
│   └── RankingHistoryRepository.java
├── analytics/
│   ├── controller/
│   │   └── AnalyticsController.java
│   ├── service/
│   │   ├── AnalyticsService.java
│   │   ├── AnalyticsServiceImpl.java
│   │   ├── ContributionGraphService.java
│   │   ├── PerformanceAnalysisService.java
│   │   ├── WeaknessAnalysisService.java
│   │   ├── StreakService.java
│   │   └── ComparisonService.java
│   ├── dto/
│   │   ├── AnalyticsSummaryResponse.java
│   │   ├── ContributionGraphResponse.java
│   │   ├── PerformanceBreakdownResponse.java
│   │   ├── TopicAnalysisResponse.java
│   │   ├── WeeklyTrendResponse.java
│   │   ├── StreakResponse.java
│   │   ├── ComparisonResponse.java
│   │   └── HeatmapResponse.java
│   ├── model/
│   │   ├── DailyActivity.java
│   │   ├── TopicPerformance.java
│   │   ├── StreakData.java
│   │   ├── WeeklyStats.java
│   │   └── PerformanceMetrics.java
│   └── repository/
│       ├── DailyActivityRepository.java
│       ├── TopicPerformanceRepository.java
│       ├── StreakDataRepository.java
│       └── WeeklyStatsRepository.java
├── dashboard/
│   ├── controller/
│   │   └── DashboardController.java
│   ├── service/
│   │   ├── DashboardService.java
│   │   ├── DashboardServiceImpl.java
│   │   ├── RecommendationService.java
│   │   ├── AIRecommendationService.java
│   │   └── PersonalizedFeedService.java
│   ├── dto/
│   │   └── DashboardOverviewResponse.java
```

```
├── RecommendationResponse.java
├── PersonalizedFeedResponse.java
├── DailyGoalResponse.java
├── QuickStatsResponse.java
├── model/
│   ├── DashboardWidget.java
│   ├── Recommendation.java
│   ├── DailyGoal.java
│   └── FeedItem.java
├── repository/
│   ├── DashboardWidgetRepository.java
│   ├── RecommendationRepository.java
│   └── DailyGoalRepository.java
├── resources/
│   ├── controller/
│   │   └── ResourceController.java
│   ├── service/
│   │   ├── ResourceService.java
│   │   ├── ResourceServiceImpl.java
│   │   ├── RoadmapService.java
│   │   ├── BlogService.java
│   │   └── InterviewPrepService.java
│   ├── dto/
│   │   ├── ResourceResponse.java
│   │   ├── RoadmapResponse.java
│   │   ├── BlogResponse.java
│   │   ├── InterviewResourceResponse.java
│   │   └── CuratedContentResponse.java
│   ├── model/
│   │   ├── Resource.java
│   │   ├── Roadmap.java
│   │   ├── RoadmapStep.java
│   │   ├── Blog.java
│   │   ├── InterviewExperience.java
│   │   └── CuratedContent.java
│   └── repository/
│       ├── ResourceRepository.java
│       ├── RoadmapRepository.java
│       ├── BlogRepository.java
│       ├── InterviewExperienceRepository.java
│       └── CuratedContentRepository.java
├── community/
│   ├── controller/
│   │   ├── CommunityController.java
│   │   ├── StudyGroupController.java
│   │   ├── DiscussionController.java
│   │   └── MentorshipController.java
│   ├── service/
│   │   ├── CommunityService.java
│   │   ├── CommunityServiceImpl.java
│   │   ├── StudyGroupService.java
│   │   ├── DiscussionService.java
│   │   ├── MentorshipService.java
│   │   ├── FollowService.java
│   │   └── NotificationService.java
│   └── dto/
```

```
├── StudyGroupRequest.java
├── StudyGroupResponse.java
├── DiscussionRequest.java
├── DiscussionResponse.java
├── MentorshipRequest.java
├── FollowRequest.java
├── NotificationResponse.java
├── model/
│   ├── StudyGroup.java
│   ├── StudyGroupMember.java
│   ├── Discussion.java
│   ├── DiscussionReply.java
│   ├── MentorshipSession.java
│   ├── UserFollow.java
│   └── Notification.java
├── repository/
│   ├── StudyGroupRepository.java
│   ├── StudyGroupMemberRepository.java
│   ├── DiscussionRepository.java
│   ├── DiscussionReplyRepository.java
│   ├── MentorshipSessionRepository.java
│   ├── UserFollowRepository.java
│   └── NotificationRepository.java
├── gamification/
│   ├── controller/
│   │   └── GamificationController.java
│   ├── service/
│   │   ├── GamificationService.java
│   │   ├── GamificationServiceImpl.java
│   │   ├── BadgeService.java
│   │   ├── XPService.java
│   │   └── LevelService.java
│   ├── dto/
│   │   ├── BadgeResponse.java
│   │   ├── XPResponse.java
│   │   ├── LevelResponse.java
│   │   └── AchievementResponse.java
│   ├── model/
│   │   ├── UserBadge.java
│   │   ├── BadgeType.java
│   │   ├── XPTransaction.java
│   │   ├── UserLevel.java
│   │   └── Achievement.java
│   └── repository/
│       ├── UserBadgeRepository.java
│       ├── BadgeTypeRepository.java
│       ├── XPTransactionRepository.java
│       ├── UserLevelRepository.java
│       └── AchievementRepository.java
├── files/
│   ├── controller/
│   │   └── FileController.java
│   ├── service/
│   │   ├── FileUploadService.java
│   │   ├── FileUploadServiceImpl.java
│   │   └── S3Service.java
```

```
├── ResumeService.java
├── ProfilePictureService.java
├── dto/
│   ├── FileUploadResponse.java
│   └── FileDownloadResponse.java
├── model/
│   ├── UploadedFile.java
│   └── FileMetadata.java
├── admin/
│   ├── controller/
│   │   ├── AdminController.java
│   │   ├── AdminProblemController.java
│   │   ├── AdminContestController.java
│   │   └── AdminUserController.java
│   ├── service/
│   │   ├── AdminService.java
│   │   ├── AdminServiceImpl.java
│   │   ├── AdminProblemService.java
│   │   ├── AdminContestService.java
│   │   └── AdminUserService.java
│   ├── dto/
│   │   ├── AdminDashboardResponse.java
│   │   ├── AdminProblemRequest.java
│   │   ├── AdminContestRequest.java
│   │   ├── AdminUserResponse.java
│   │   └── AdminStatsResponse.java
│   └── model/
│       ├── AdminLog.java
│       └── SystemMetrics.java
├── websocket/
│   ├── controller/
│   │   └── WebSocketController.java
│   ├── service/
│   │   ├── WebSocketService.java
│   │   ├── LiveContestService.java
│   │   └── NotificationService.java
│   ├── config/
│   │   └── WebSocketConfig.java
│   └── handler/
│       ├── ContestWebSocketHandler.java
│       └── NotificationWebSocketHandler.java
├── external/
│   ├── service/
│   │   ├── LeetCodeApiService.java
│   │   ├── CodeforcesApiService.java
│   │   ├── CodeChefApiService.java
│   │   ├── GeeksforGeeksApiService.java
│   │   └── ExternalPlatformSyncService.java
│   ├── dto/
│   │   ├── LeetCodeProfileResponse.java
│   │   ├── CodeforcesProfileResponse.java
│   │   ├── CodeChefProfileResponse.java
│   │   └── ExternalSubmissionResponse.java
│   └── model/
│       ├── ExternalProfile.java
│       └── ExternalSubmission.java
```



```
└─ common/
    ├── dto/
    │   ├── ApiResponse.java
    │   ├── PageResponse.java
    │   ├── ErrorResponse.java
    │   └── BaseDto.java
    ├── exception/
    │   ├── GlobalExceptionHandler.java
    │   ├── ResourceNotFoundException.java
    │   ├── UnauthorizedException.java
    │   ├── ValidationException.java
    │   ├── CodeExecutionException.java
    │   └── ExternalApiException.java
    ├── utils/
    │   ├── DateUtils.java
    │   ├── StringUtils.java
    │   ├── ValidationUtils.java
    │   ├── SecurityUtils.java
    │   ├── FileUtils.java
    │   ├── CodeExecutionUtils.java
    │   └── RankingUtils.java
    ├── constants/
    │   ├── AppConstants.java
    │   ├── SecurityConstants.java
    │   ├── CacheConstants.java
    │   ├── FileConstants.java
    │   └── ApiConstants.java
    ├── enums/
    │   ├── UserRole.java
    │   ├── ProblemDifficulty.java
    │   ├── SubmissionStatus.java
    │   ├── ContestType.java
    │   ├── Language.java
    │   ├── BadgeType.java
    │   └── NotificationType.java
    ├── validators/
    │   ├── EmailValidator.java
    │   ├── PrnValidator.java
    │   ├── UrlValidator.java
    │   └── PasswordValidator.java
    └── aspects/
        ├── LoggingAspect.java
        ├── PerformanceAspect.java
        ├── SecurityAspect.java
        └── CachingAspect.java
resources/
├── application.yml
├── application-dev.yml
├── application-prod.yml
├── application-test.yml
├── db/
│   └── migration/
│       ├── V1__Create_user_tables.sql
│       ├── V2__Create_problem_tables.sql
│       ├── V3__Create_contest_tables.sql
│       └── V4__Create_analytics_tables.sql
```

```
├── V5__Create_community_tables.sql
├── V6__Create_admin_tables.sql
├── static/
│   ├── images/
│   ├── css/
│   ├── js/
│   └── templates/
│       ├── email/
│       │   ├── welcome.html
│       │   ├── password-reset.html
│       │   └── contest-reminder.html
│       └── pdf/
│           ├── certificate.html
│           └── progress-report.html
├── docker/
│   ├── Dockerfile
│   ├── docker-compose.yml
│   ├── docker-compose.dev.yml
│   ├── docker-compose.prod.yml
│   └── code-execution/
│       ├── Dockerfile.java
│       ├── Dockerfile.python
│       ├── Dockerfile.cpp
│       ├── Dockerfile.javascript
│       └── sandbox-config.yml
├── scripts/
│   ├── start-dev.sh
│   ├── start-prod.sh
│   ├── deploy.sh
│   ├── backup-db.sh
│   └── setup-environment.sh
├── docs/
│   ├── api/
│   │   ├── swagger.yml
│   │   ├── endpoints.md
│   │   └── authentication.md
│   ├── deployment/
│   │   ├── aws-setup.md
│   │   ├── docker-setup.md
│   │   └── kubernetes.md
│   └── development/
│       ├── setup.md
│       ├── contributing.md
│       └── testing.md
├── tests/
│   ├── unit/
│   ├── integration/
│   └── e2e/
├── pom.xml
├── mvnw
├── mvnw.cmd
├── .gitignore
├── README.md
└── LICENSE
```

Frontend Architecture (React + TypeScript)

```
frontend/
├── public/
│   ├── index.html
│   ├── favicon.ico
│   ├── manifest.json
│   ├── robots.txt
│   ├── images/
│   │   ├── logo.svg
│   │   ├── default-avatar.png
│   │   ├── badges/
│   │   │   ├── bronze.svg
│   │   │   ├── silver.svg
│   │   │   ├── gold.svg
│   │   │   └── platinum.svg
│   │   ├── platforms/
│   │   │   ├── leetcode.svg
│   │   │   ├── codeforces.svg
│   │   │   ├── codechef.svg
│   │   │   └── gfg.svg
│   │   └── illustrations/
│   │       ├── welcome.svg
│   │       ├── coding.svg
│   │       └── achievement.svg
│   └── service-worker.js
├── src/
│   ├── index.tsx
│   ├── App.tsx
│   ├── App.css
│   ├── index.css
│   ├── setupTests.ts
│   ├── reportWebVitals.ts
│   ├── react-app-env.d.ts
│   ├── assets/
│   │   ├── images/
│   │   │   ├── index.ts
│   │   │   ├── logos/
│   │   │   ├── icons/
│   │   │   ├── backgrounds/
│   │   │   └── illustrations/
│   │   ├── videos/
│   │   │   ├── hero-background.mp4
│   │   │   └── tutorial-intro.mp4
│   │   ├── audio/
│   │   │   ├── notification.mp3
│   │   │   └── success.mp3
│   │   └── fonts/
│   │       ├── Inter/
│   │       ├── JetBrainsMono/
│   │       └── index.css
│   ├── components/
│   │   ├── common/
│   │   │   ├── ui/
│   │   │   │   ├── Button/
│   │   │   │   └── index.ts
```

- ├── Button.tsx
- ├── Button.test.tsx
- ├── Button.stories.tsx
- ├── Button.module.css
- ├── types.ts
- ├── Input/
 - ├── index.ts
 - ├── Input.tsx
 - ├── Input.test.tsx
 - ├── Input.module.css
 - ├── types.ts
- ├── Select/
 - ├── index.ts
 - ├── Select.tsx
 - ├── Select.test.tsx
 - ├── Select.module.css
 - ├── types.ts
- ├── MultiSelect/
 - ├── index.ts
 - ├── MultiSelect.tsx
 - ├── MultiSelect.test.tsx
 - ├── MultiSelect.module.css
 - ├── types.ts
- ├── Modal/
 - ├── index.ts
 - ├── Modal.tsx
 - ├── Modal.test.tsx
 - ├── Modal.module.css
 - ├── types.ts
- ├── Avatar/
 - ├── index.ts
 - ├── Avatar.tsx
 - ├── Avatar.test.tsx
 - ├── Avatar.module.css
 - ├── types.ts
- ├── Badge/
 - ├── index.ts
 - ├── Badge.tsx
 - ├── Badge.test.tsx
 - ├── Badge.module.css
 - ├── types.ts
- ├── Card/
 - ├── index.ts
 - ├── Card.tsx
 - ├── Card.test.tsx
 - ├── Card.module.css
 - ├── types.ts
- ├── Spinner/
 - ├── index.ts
 - ├── Spinner.tsx
 - ├── Spinner.test.tsx
 - ├── Spinner.module.css
 - ├── types.ts
- ├── Tooltip/
 - ├── index.ts
 - ├── Tooltip.tsx

```
├── Tooltip.test.tsx
├── Tooltip.module.css
├── types.ts
├── Dropdown/
│   ├── index.ts
│   ├── Dropdown.tsx
│   ├── Dropdown.test.tsx
│   ├── Dropdown.module.css
│   └── types.ts
├── Tabs/
│   ├── index.ts
│   ├── Tabs.tsx
│   ├── Tabs.test.tsx
│   ├── Tabs.module.css
│   └── types.ts
├── Pagination/
│   ├── index.ts
│   ├── Pagination.tsx
│   ├── Pagination.test.tsx
│   ├── Pagination.module.css
│   └── types.ts
├── ProgressBar/
│   ├── index.ts
│   ├── ProgressBar.tsx
│   ├── ProgressBar.test.tsx
│   ├── ProgressBar.module.css
│   └── types.ts
├── Toast/
│   ├── index.ts
│   ├── Toast.tsx
│   ├── ToastContainer.tsx
│   ├── Toast.test.tsx
│   ├── Toast.module.css
│   └── types.ts
├── index.ts
├── forms/
│   ├── FormField/
│   │   ├── index.ts
│   │   ├── FormField.tsx
│   │   ├── FormField.test.tsx
│   │   ├── FormField.module.css
│   │   └── types.ts
│   ├── FormValidator/
│   │   ├── index.ts
│   │   ├── FormValidator.tsx
│   │   ├── FormValidator.test.tsx
│   │   └── types.ts
│   ├── FileUpload/
│   │   ├── index.ts
│   │   ├── FileUpload.tsx
│   │   ├── FileUpload.test.tsx
│   │   ├── FileUpload.module.css
│   │   └── types.ts
│   └── index.ts
├── navigation/
├── Breadcrumb/
```

```
├── index.ts
├── Breadcrumb.tsx
├── Breadcrumb.test.tsx
├── Breadcrumb.module.css
├── types.ts
├── BackButton/
│   ├── index.ts
│   ├── BackButton.tsx
│   ├── BackButton.test.tsx
│   ├── BackButton.module.css
│   └── types.ts
├── index.ts
├── layout/
│   ├── Header/
│   │   ├── index.ts
│   │   ├── Header.tsx
│   │   ├── Header.test.tsx
│   │   ├── Header.module.css
│   │   ├── types.ts
│   │   └── components/
│   │       ├── UserMenu/
│   │       │   ├── index.ts
│   │       │   ├── UserMenu.tsx
│   │       │   ├── UserMenu.test.tsx
│   │       │   ├── UserMenu.module.css
│   │       │   └── types.ts
│   │       ├── NotificationBell/
│   │       │   ├── index.ts
│   │       │   ├── NotificationBell.tsx
│   │       │   ├── NotificationBell.test.tsx
│   │       │   ├── NotificationBell.module.css
│   │       │   └── types.ts
│   │       └── SearchBar/
│   │           ├── index.ts
│   │           ├── SearchBar.tsx
│   │           ├── SearchBar.test.tsx
│   │           ├── SearchBar.module.css
│   │           └── types.ts
│   └── Sidebar/
│       ├── index.ts
│       ├── Sidebar.tsx
│       ├── Sidebar.test.tsx
│       ├── Sidebar.module.css
│       ├── types.ts
│       └── components/
│           ├── NavigationItem/
│           │   ├── index.ts
│           │   ├── NavigationItem.tsx
│           │   ├── NavigationItem.test.tsx
│           │   ├── NavigationItem.module.css
│           │   └── types.ts
│           └── ProfileSummary/
│               ├── index.ts
│               ├── ProfileSummary.tsx
│               ├── ProfileSummary.test.tsx
│               └── ProfileSummary.module.css
```

```
├── types.ts
├── DailyGoalWidget/
│   ├── index.ts
│   ├── DailyGoalWidget.tsx
│   ├── DailyGoalWidget.test.tsx
│   ├── DailyGoalWidget.module.css
│   └── types.ts
├── Footer/
│   ├── index.ts
│   ├── Footer.tsx
│   ├── Footer.test.tsx
│   ├── Footer.module.css
│   └── types.ts
├── MainLayout/
│   ├── index.ts
│   ├── MainLayout.tsx
│   ├── MainLayout.test.tsx
│   ├── MainLayout.module.css
│   └── types.ts
├── AuthLayout/
│   ├── index.ts
│   ├── AuthLayout.tsx
│   ├── AuthLayout.test.tsx
│   ├── AuthLayout.module.css
│   └── types.ts
├── index.ts
├── features/
│   ├── auth/
│   │   ├── components/
│   │   │   ├── LoginForm/
│   │   │   │   ├── index.ts
│   │   │   │   ├── LoginForm.tsx
│   │   │   │   ├── LoginForm.test.tsx
│   │   │   │   ├── LoginForm.module.css
│   │   │   │   └── types.ts
│   │   │   ├── RegisterForm/
│   │   │   │   ├── index.ts
│   │   │   │   ├── RegisterForm.tsx
│   │   │   │   ├── RegisterForm.test.tsx
│   │   │   │   ├── RegisterForm.module.css
│   │   │   │   └── types.ts
│   │   │   ├── ForgotPasswordForm/
│   │   │   │   ├── index.ts
│   │   │   │   ├── ForgotPasswordForm.tsx
│   │   │   │   ├── ForgotPasswordForm.test.tsx
│   │   │   │   ├── ForgotPasswordForm.module.css
│   │   │   │   └── types.ts
│   │   │   ├── GoogleAuthButton/
│   │   │   │   ├── index.ts
│   │   │   │   ├── GoogleAuthButton.tsx
│   │   │   │   ├── GoogleAuthButton.test.tsx
│   │   │   │   ├── GoogleAuthButton.module.css
│   │   │   │   └── types.ts
│   │   │   └── PrivateRoute/
│   │   │       ├── index.ts
│   │   │       └── PrivateRoute.tsx
```

```
└─ PrivateRoute.test.tsx
└─ types.ts
└─ pages/
  └─ LoginPage/
    └─ index.ts
    └─ LoginPage.tsx
    └─ LoginPage.test.tsx
    └─ LoginPage.module.css
    └─ types.ts
  └─ RegisterPage/
    └─ index.ts
    └─ RegisterPage.tsx
    └─ RegisterPage.test.tsx
    └─ RegisterPage.module.css
    └─ types.ts
  └─ ForgotPasswordPage/
    └─ index.ts
    └─ ForgotPasswordPage.tsx
    └─ ForgotPasswordPage.test.tsx
    └─ ForgotPasswordPage.module.css
    └─ types.ts
  └─ ResetPasswordPage/
    └─ index.ts
    └─ ResetPasswordPage.tsx
    └─ ResetPasswordPage.test.tsx
    └─ ResetPasswordPage.module.css
    └─ types.ts
└─ hooks/
  └─ useAuth.ts
  └─ useLogin.ts
  └─ useRegister.ts
  └─ useGoogleAuth.ts
  └─ useLogout.ts
  └─ usePasswordReset.ts
  └─ index.ts
└─ services/
  └─ authApi.ts
  └─ authHelpers.ts
  └─ tokenService.ts
  └─ googleAuthService.ts
  └─ index.ts
└─ store/
  └─ authSlice.ts
  └─ authSelectors.ts
  └─ authThunks.ts
  └─ index.ts
└─ types/
  └─ auth.types.ts
  └─ user.types.ts
  └─ index.ts
└─ utils/
  └─ authValidators.ts
  └─ authConstants.ts
  └─ index.ts
└─ index.ts
└─ onboarding/
```



```
— components/
  — OnboardingForm/
    — index.ts
    — OnboardingForm.tsx
    — OnboardingForm.test.tsx
    — OnboardingForm.module.css
    — types.ts
  — StepIndicator/
    — index.ts
    — StepIndicator.tsx
    — StepIndicator.test.tsx
    — StepIndicator.module.css
    — types.ts
  — PersonalInfoStep/
    — index.ts
    — PersonalInfoStep.tsx
    — PersonalInfoStep.test.tsx
    — PersonalInfoStep.module.css
    — types.ts
  — AcademicInfoStep/
    — index.ts
    — AcademicInfoStep.tsx
    — AcademicInfoStep.test.tsx
    — AcademicInfoStep.module.css
    — types.ts
  — CompetitiveProgrammingStep/
    — index.ts
    — CompetitiveProgrammingStep.tsx
    — CompetitiveProgrammingStep.test.tsx
    — CompetitiveProgrammingStep.module.css
    — types.ts
  — PreferencesStep/
    — index.ts
    — PreferencesStep.tsx
    — PreferencesStep.test.tsx
    — PreferencesStep.module.css
    — types.ts
  — FilesUploadStep/
    — index.ts
    — FilesUploadStep.tsx
    — FilesUploadStep.test.tsx
    — FilesUploadStep.module.css
    — types.ts
— pages/
  — OnboardingPage/
    — index.ts
    — OnboardingPage.tsx
    — OnboardingPage.test.tsx
    — OnboardingPage.module.css
    — types.ts
— hooks/
  — useOnboarding.ts
  — useOnboardingForm.ts
  — useFileUpload.ts
  — index.ts
— services/
```

```

├── onboardingApi.ts
├── fileUploadService.ts
├── index.ts
├── store/
│   ├── onboardingSlice.ts
│   ├── onboardingSelectors.ts
│   └── index.ts
├── types/
│   ├── onboarding.types.ts
│   └── index.ts
├── utils/
│   ├── onboardingValidators.ts
│   ├── onboardingConstants.ts
│   └── index.ts
├── index.ts
├── dashboard/
│   ├── components/
│   │   ├── DashboardOverview/
│   │   │   ├── index.ts
│   │   │   ├── DashboardOverview.tsx
│   │   │   ├── DashboardOverview.test.tsx
│   │   │   ├── DashboardOverview.module.css
│   │   │   └── types.ts
│   │   ├── OverviewCards/
│   │   │   ├── index.ts
│   │   │   ├── OverviewCards.tsx
│   │   │   ├── OverviewCards.test.tsx
│   │   │   ├── OverviewCards.module.css
│   │   │   └── types.ts
│   │   ├── DailyGoalProgress/
│   │   │   ├── index.ts
│   │   │   ├── DailyGoalProgress.tsx
│   │   │   ├── DailyGoalProgress.test.tsx
│   │   │   ├── DailyGoalProgress.module.css
│   │   │   └── types.ts
│   │   ├── RecommendationFeed/
│   │   │   ├── index.ts
│   │   │   ├── RecommendationFeed.tsx
│   │   │   ├── RecommendationFeed.test.tsx
│   │   │   ├── RecommendationFeed.module.css
│   │   │   └── types.ts
│   │   ├── ExploreSection/
│   │   │   ├── index.ts
│   │   │   ├── ExploreSection.tsx
│   │   │   ├── ExploreSection.test.tsx
│   │   │   ├── ExploreSection.module.css
│   │   │   └── types.ts
│   │   ├── UpcomingContests/
│   │   │   ├── index.ts
│   │   │   ├── UpcomingContests.tsx
│   │   │   ├── UpcomingContests.test.tsx
│   │   │   ├── UpcomingContests.module.css
│   │   │   └── types.ts
│   │   └── RecentActivity/
│   │       ├── index.ts
│   │       └── RecentActivity.tsx

```

```
├── RecentActivity.test.tsx
├── RecentActivity.module.css
├── types.ts
├── pages/
│   ├── DashboardPage/
│   │   ├── index.ts
│   │   ├── DashboardPage.tsx
│   │   ├── DashboardPage.test.tsx
│   │   ├── DashboardPage.module.css
│   │   └── types.ts
│   ├── hooks/
│   │   ├── useDashboard.ts
│   │   ├── useRecommendations.ts
│   │   ├── useDailyGoal.ts
│   │   └── index.ts
│   ├── services/
│   │   ├── dashboardApi.ts
│   │   ├── recommendationService.ts
│   │   └── index.ts
│   ├── store/
│   │   ├── dashboardSlice.ts
│   │   ├── dashboardSelectors.ts
│   │   └── index.ts
│   ├── types/
│   │   ├── dashboard.types.ts
│   │   └── index.ts
│   └── index.ts
├── profile/
│   ├── components/
│   │   ├── ProfileHeader/
│   │   │   ├── index.ts
│   │   │   ├── ProfileHeader.tsx
│   │   │   ├── ProfileHeader.test.tsx
│   │   │   ├── ProfileHeader.module.css
│   │   │   └── types.ts
│   │   ├── BasicInfoSection/
│   │   │   ├── index.ts
│   │   │   ├── BasicInfoSection.tsx
│   │   │   ├── BasicInfoSection.test.tsx
│   │   │   ├── BasicInfoSection.module.css
│   │   │   └── types.ts
│   │   ├── CompetitiveProfilesSection/
│   │   │   ├── index.ts
│   │   │   ├── CompetitiveProfilesSection.tsx
│   │   │   ├── CompetitiveProfilesSection.test.tsx
│   │   │   ├── CompetitiveProfilesSection.module.css
│   │   │   └── types.ts
│   │   ├── ContributionGraph/
│   │   │   ├── index.ts
│   │   │   ├── ContributionGraph.tsx
│   │   │   ├── ContributionGraph.test.tsx
│   │   │   ├── ContributionGraph.module.css
│   │   │   └── types.ts
│   │   └── RecentActivitySection/
│   │       ├── index.ts
│   │       └── RecentActivitySection.tsx
```

```
├── RecentActivitySection.test.tsx
├── RecentActivitySection.module.css
├── types.ts
├── StatsSection/
│   ├── index.ts
│   ├── StatsSection.tsx
│   ├── StatsSection.test.tsx
│   ├── StatsSection.module.css
│   └── types.ts
├── EditProfileForm/
│   ├── index.ts
│   ├── EditProfileForm.tsx
│   ├── EditProfileForm.test.tsx
│   ├── EditProfileForm.module.css
│   └── types.ts
├── ResumeDownload/
│   ├── index.ts
│   ├── ResumeDownload.tsx
│   ├── ResumeDownload.test.tsx
│   ├── ResumeDownload.module.css
│   └── types.ts
├── pages/
│   ├── ProfilePage/
│   │   ├── index.ts
│   │   ├── ProfilePage.tsx
│   │   ├── ProfilePage.test.tsx
│   │   ├── ProfilePage.module.css
│   │   └── types.ts
│   └── EditProfilePage/
│       ├── index.ts
│       ├── EditProfilePage.tsx
│       ├── EditProfilePage.test.tsx
│       ├── EditProfilePage.module.css
│       └── types.ts
├── hooks/
│   ├── useProfile.ts
│   ├── useContributionGraph.ts
│   ├── useProfileEdit.ts
│   ├── useResumeDownload.ts
│   └── index.ts
├── services/
│   ├── profileApi.ts
│   ├── contributionService.ts
│   └── index.ts
├── store/
│   ├── profileSlice.ts
│   ├── profileSelectors.ts
│   └── index.ts
├── types/
│   ├── profile.types.ts
│   └── index.ts
├── index.ts
├── problems/
│   ├── components/
│   │   ├── ProblemList/
│   │   │   └── index.ts
```

```
├── ProblemList.tsx
├── ProblemList.test.tsx
├── ProblemList.module.css
├── types.ts
├── ProblemCard/
│   ├── index.ts
│   ├── ProblemCard.tsx
│   ├── ProblemCard.test.tsx
│   ├── ProblemCard.module.css
│   └── types.ts
├── ProblemDetails/
│   ├── index.ts
│   ├── ProblemDetails.tsx
│   ├── ProblemDetails.test.tsx
│   ├── ProblemDetails.module.css
│   └── types.ts
├── CodeEditor/
│   ├── index.ts
│   ├── CodeEditor.tsx
│   ├── CodeEditor.test.tsx
│   ├── CodeEditor.module.css
│   ├── types.ts
│   └── components/
│       ├── EditorHeader/
│           ├── index.ts
│           ├── EditorHeader.tsx
│           ├── EditorHeader.test.tsx
│           ├── EditorHeader.module.css
│           └── types.ts
│       ├── LanguageSelector/
│           ├── index.ts
│           ├── LanguageSelector.tsx
│           ├── LanguageSelector.test.tsx
│           ├── LanguageSelector.module.css
│           └── types.ts
│       ├── ThemeSelector/
│           ├── index.ts
│           ├── ThemeSelector.tsx
│           ├── ThemeSelector.test.tsx
│           ├── ThemeSelector.module.css
│           └── types.ts
│       └── EditorActions/
│           ├── index.ts
│           ├── EditorActions.tsx
│           ├── EditorActions.test.tsx
│           ├── EditorActions.module.css
│           └── types.ts
├── TestCasePanel/
│   ├── index.ts
│   ├── TestCasePanel.tsx
│   ├── TestCasePanel.test.tsx
│   ├── TestCasePanel.module.css
│   └── types.ts
├── ExecutionResult/
│   ├── index.ts
│   └── ExecutionResult.tsx
```

```
├── ExecutionResult.test.tsx
├── ExecutionResult.module.css
├── types.ts
├── SubmissionHistory/
│   ├── index.ts
│   ├── SubmissionHistory.tsx
│   ├── SubmissionHistory.test.tsx
│   ├── SubmissionHistory.module.css
│   └── types.ts
├── ProblemFilters/
│   ├── index.ts
│   ├── ProblemFilters.tsx
│   ├── ProblemFilters.test.tsx
│   ├── ProblemFilters.module.css
│   └── types.ts
├── TestPlayground/
│   ├── index.ts
│   ├── TestPlayground.tsx
│   ├── TestPlayground.test.tsx
│   ├── TestPlayground.module.css
│   └── types.ts
├── DiscussionSection/
│   ├── index.ts
│   ├── DiscussionSection.tsx
│   ├── DiscussionSection.test.tsx
│   ├── DiscussionSection.module.css
│   └── types.ts
├── pages/
│   ├── ProblemsPage/
│   │   ├── index.ts
│   │   ├── ProblemsPage.tsx
│   │   ├── ProblemsPage.test.tsx
│   │   ├── ProblemsPage.module.css
│   │   └── types.ts
│   ├── ProblemDetailPage/
│   │   ├── index.ts
│   │   ├── ProblemDetailPage.tsx
│   │   ├── ProblemDetailPage.test.tsx
│   │   ├── ProblemDetailPage.module.css
│   │   └── types.ts
│   └── PlaygroundPage/
│       ├── index.ts
│       ├── PlaygroundPage.tsx
│       ├── PlaygroundPage.test.tsx
│       ├── PlaygroundPage.module.css
│       └── types.ts
├── hooks/
│   ├── useProblems.ts
│   ├── useProblemDetail.ts
│   ├── useCodeEditor.ts
│   ├── useCodeExecution.ts
│   ├── useProblemFilters.ts
│   ├── useSubmissionHistory.ts
│   └── index.ts
├── services/
└── problemsApi.ts
```

```
├── codeExecutionService.ts
├── submissionService.ts
├── index.ts
├── store/
│   ├── problemsSlice.ts
│   ├── problemsSelectors.ts
│   ├── codeEditorSlice.ts
│   ├── submissionSlice.ts
│   └── index.ts
├── types/
│   ├── problem.types.ts
│   ├── submission.types.ts
│   ├── codeEditor.types.ts
│   └── index.ts
├── utils/
│   ├── codeTemplates.ts
│   ├── languageConfigs.ts
│   ├── editorThemes.ts
│   └── index.ts
├── index.ts
├── sheets/
│   ├── components/
│   │   ├── SheetList/
│   │   │   ├── index.ts
│   │   │   ├── SheetList.tsx
│   │   │   ├── SheetList.test.tsx
│   │   │   ├── SheetList.module.css
│   │   │   └── types.ts
│   │   ├── SheetCard/
│   │   │   ├── index.ts
│   │   │   ├── SheetCard.tsx
│   │   │   ├── SheetCard.test.tsx
│   │   │   ├── SheetCard.module.css
│   │   │   └── types.ts
│   │   ├── SheetProblemList/
│   │   │   ├── index.ts
│   │   │   ├── SheetProblemList.tsx
│   │   │   ├── SheetProblemList.test.tsx
│   │   │   ├── SheetProblemList.module.css
│   │   │   └── types.ts
│   │   ├── ProgressTracker/
│   │   │   ├── index.ts
│   │   │   ├── ProgressTracker.tsx
│   │   │   ├── ProgressTracker.test.tsx
│   │   │   ├── ProgressTracker.module.css
│   │   │   └── types.ts
│   │   ├── BookmarkButton/
│   │   │   ├── index.ts
│   │   │   ├── BookmarkButton.tsx
│   │   │   ├── BookmarkButton.test.tsx
│   │   │   ├── BookmarkButton.module.css
│   │   │   └── types.ts
│   │   └── CustomSheetBuilder/
│   │       ├── index.ts
│   │       ├── CustomSheetBuilder.tsx
│   │       └── CustomSheetBuilder.test.tsx
```

```
├── CustomSheetBuilder.module.css
├── types.ts
├── pages/
│   ├── SheetsPage/
│   │   ├── index.ts
│   │   ├── SheetsPage.tsx
│   │   ├── SheetsPage.test.tsx
│   │   ├── SheetsPage.module.css
│   │   └── types.ts
│   ├── SheetDetailPage/
│   │   ├── index.ts
│   │   ├── SheetDetailPage.tsx
│   │   ├── SheetDetailPage.test.tsx
│   │   ├── SheetDetailPage.module.css
│   │   └── types.ts
│   └── BookmarksPage/
│       ├── index.ts
│       ├── BookmarksPage.tsx
│       ├── BookmarksPage.test.tsx
│       ├── BookmarksPage.module.css
│       └── types.ts
├── hooks/
│   ├── useSheets.ts
│   ├── useSheetDetail.ts
│   ├── useBookmarks.ts
│   ├── useSheetProgress.ts
│   └── index.ts
├── services/
│   ├── sheetsApi.ts
│   ├── bookmarkService.ts
│   └── index.ts
├── store/
│   ├── sheetsSlice.ts
│   ├── sheetsSelectors.ts
│   └── index.ts
├── types/
│   ├── sheet.types.ts
│   └── index.ts
├── index.ts
├── contests/
│   ├── components/
│   │   ├── ContestList/
│   │   │   ├── index.ts
│   │   │   ├── ContestList.tsx
│   │   │   ├── ContestList.test.tsx
│   │   │   ├── ContestList.module.css
│   │   │   └── types.ts
│   │   ├── ContestCard/
│   │   │   ├── index.ts
│   │   │   ├── ContestCard.tsx
│   │   │   ├── ContestCard.test.tsx
│   │   │   ├── ContestCard.module.css
│   │   │   └── types.ts
│   │   └── ContestHeader/
│   │       ├── index.ts
│   │       └── ContestHeader.tsx
```



```
├── ContestHeader.test.tsx
├── ContestHeader.module.css
├── types.ts
├── ContestTimer/
│   ├── index.ts
│   ├── ContestTimer.tsx
│   ├── ContestTimer.test.tsx
│   ├── ContestTimer.module.css
│   └── types.ts
├── LiveContestInterface/
│   ├── index.ts
│   ├── LiveContestInterface.tsx
│   ├── LiveContestInterface.test.tsx
│   ├── LiveContestInterface.module.css
│   └── types.ts
├── ContestRankings/
│   ├── index.ts
│   ├── ContestRankings.tsx
│   ├── ContestRankings.test.tsx
│   ├── ContestRankings.module.css
│   └── types.ts
├── PostContestAnalysis/
│   ├── index.ts
│   ├── PostContestAnalysis.tsx
│   ├── PostContestAnalysis.test.tsx
│   ├── PostContestAnalysis.module.css
│   └── types.ts
├── pages/
│   ├── ContestsPage/
│   │   ├── index.ts
│   │   ├── ContestsPage.tsx
│   │   ├── ContestsPage.test.tsx
│   │   ├── ContestsPage.module.css
│   │   └── types.ts
│   ├── ContestDetailPage/
│   │   ├── index.ts
│   │   ├── ContestDetailPage.tsx
│   │   ├── ContestDetailPage.test.tsx
│   │   ├── ContestDetailPage.module.css
│   │   └── types.ts
│   └── LiveContestPage/
│       ├── index.ts
│       ├── LiveContestPage.tsx
│       ├── LiveContestPage.test.tsx
│       ├── LiveContestPage.module.css
│       └── types.ts
├── hooks/
│   ├── useContests.ts
│   ├── useContestDetail.ts
│   ├── useLiveContest.ts
│   ├── useContestTimer.ts
│   ├── useContestRankings.ts
│   └── index.ts
├── services/
│   ├── contestsApi.ts
│   └── liveContestService.ts
```

```
├── contestWebSocketService.ts
├── index.ts
├── store/
│   ├── contestsSlice.ts
│   ├── contestsSelectors.ts
│   ├── liveContestSlice.ts
│   └── index.ts
├── types/
│   ├── contest.types.ts
│   ├── liveContest.types.ts
│   └── index.ts
└── index.ts
├── leaderboard/
│   ├── components/
│   │   ├── GlobalLeaderboard/
│   │   │   ├── index.ts
│   │   │   ├── GlobalLeaderboard.tsx
│   │   │   ├── GlobalLeaderboard.test.tsx
│   │   │   ├── GlobalLeaderboard.module.css
│   │   │   └── types.ts
│   │   ├── CollegeLeaderboard/
│   │   │   ├── index.ts
│   │   │   ├── CollegeLeaderboard.tsx
│   │   │   ├── CollegeLeaderboard.test.tsx
│   │   │   ├── CollegeLeaderboard.module.css
│   │   │   └── types.ts
│   │   ├── FriendsLeaderboard/
│   │   │   ├── index.ts
│   │   │   ├── FriendsLeaderboard.tsx
│   │   │   ├── FriendsLeaderboard.test.tsx
│   │   │   ├── FriendsLeaderboard.module.css
│   │   │   └── types.ts
│   │   ├── LeaderboardEntry/
│   │   │   ├── index.ts
│   │   │   ├── LeaderboardEntry.tsx
│   │   │   ├── LeaderboardEntry.test.tsx
│   │   │   ├── LeaderboardEntry.module.css
│   │   │   └── types.ts
│   │   ├── LeaderboardFilters/
│   │   │   ├── index.ts
│   │   │   ├── LeaderboardFilters.tsx
│   │   │   ├── LeaderboardFilters.test.tsx
│   │   │   ├── LeaderboardFilters.module.css
│   │   │   └── types.ts
│   │   └── RankingBadge/
│   │       ├── index.ts
│   │       ├── RankingBadge.tsx
│   │       ├── RankingBadge.test.tsx
│   │       ├── RankingBadge.module.css
│   │       └── types.ts
│   └── pages/
│       └── LeaderboardPage/
│           ├── index.ts
│           ├── LeaderboardPage.tsx
│           ├── LeaderboardPage.test.tsx
│           └── LeaderboardPage.module.css
```

```
└─ types.ts
└─ hooks/
  └─ useLeaderboard.ts
  └─ useGlobalRankings.ts
  └─ useCollegeRankings.ts
  └─ useFriendsRankings.ts
  └─ index.ts
└─ services/
  └─ leaderboardApi.ts
  └─ index.ts
└─ store/
  └─ leaderboardSlice.ts
  └─ leaderboardSelectors.ts
  └─ index.ts
└─ types/
  └─ leaderboard.types.ts
  └─ index.ts
└─ index.ts
└─ analytics/
  └─ components/
    └─ AnalyticsSummary/
      └─ index.ts
      └─ AnalyticsSummary.tsx
      └─ AnalyticsSummary.test.tsx
      └─ AnalyticsSummary.module.css
      └─ types.ts
    └─ WeeklyProgressChart/
      └─ index.ts
      └─ WeeklyProgressChart.tsx
      └─ WeeklyProgressChart.test.tsx
      └─ WeeklyProgressChart.module.css
      └─ types.ts
    └─ TopicPieChart/
      └─ index.ts
      └─ TopicPieChart.tsx
      └─ TopicPieChart.test.tsx
      └─ TopicPieChart.module.css
      └─ types.ts
    └─ PerformanceHeatmap/
      └─ index.ts
      └─ PerformanceHeatmap.tsx
      └─ PerformanceHeatmap.test.tsx
      └─ PerformanceHeatmap.module.css
      └─ types.ts
    └─ WeaknessAnalysis/
      └─ index.ts
      └─ WeaknessAnalysis.tsx
      └─ WeaknessAnalysis.test.tsx
      └─ WeaknessAnalysis.module.css
      └─ types.ts
    └─ CompareWithPeers/
      └─ index.ts
      └─ CompareWithPeers.tsx
      └─ CompareWithPeers.test.tsx
      └─ CompareWithPeers.module.css
      └─ types.ts
```

```
├── StreakTracker/
│   ├── index.ts
│   ├── StreakTracker.tsx
│   ├── StreakTracker.test.tsx
│   ├── StreakTracker.module.css
│   └── types.ts
├── TimeAnalysis/
│   ├── index.ts
│   ├── TimeAnalysis.tsx
│   ├── TimeAnalysis.test.tsx
│   ├── TimeAnalysis.module.css
│   └── types.ts
├── pages/
│   ├── AnalyticsPage/
│   │   ├── index.ts
│   │   ├── AnalyticsPage.tsx
│   │   ├── AnalyticsPage.test.tsx
│   │   ├── AnalyticsPage.module.css
│   │   └── types.ts
│   ├── hooks/
│   │   ├── useAnalytics.ts
│   │   ├── useWeeklyProgress.ts
│   │   ├── useTopicAnalysis.ts
│   │   ├── useWeaknessAnalysis.ts
│   │   ├── useCompareWithPeers.ts
│   │   ├── useStreakData.ts
│   │   └── index.ts
│   ├── services/
│   │   ├── analyticsApi.ts
│   │   └── index.ts
│   ├── store/
│   │   ├── analyticsSlice.ts
│   │   ├── analyticsSelectors.ts
│   │   └── index.ts
│   ├── types/
│   │   ├── analytics.types.ts
│   │   └── index.ts
│   └── index.ts
├── resources/
│   ├── components/
│   │   ├── RoadmapViewer/
│   │   │   ├── index.ts
│   │   │   ├── RoadmapViewer.tsx
│   │   │   ├── RoadmapViewer.test.tsx
│   │   │   ├── RoadmapViewer.module.css
│   │   │   └── types.ts
│   │   ├── ResourceCard/
│   │   │   ├── index.ts
│   │   │   ├── ResourceCard.tsx
│   │   │   ├── ResourceCard.test.tsx
│   │   │   ├── ResourceCard.module.css
│   │   │   └── types.ts
│   │   └── BlogList/
│   │       ├── index.ts
│   │       ├── BlogList.tsx
│   │       └── BlogList.test.tsx
```

```
├── BlogList.module.css
├── types.ts
├── InterviewPrepSection/
│   ├── index.ts
│   ├── InterviewPrepSection.tsx
│   ├── InterviewPrepSection.test.tsx
│   ├── InterviewPrepSection.module.css
│   └── types.ts
├── CompanyExperience/
│   ├── index.ts
│   ├── CompanyExperience.tsx
│   ├── CompanyExperience.test.tsx
│   ├── CompanyExperience.module.css
│   └── types.ts
├── ResumeBuilder/
│   ├── index.ts
│   ├── ResumeBuilder.tsx
│   ├── ResumeBuilder.test.tsx
│   ├── ResumeBuilder.module.css
│   └── types.ts
├── pages/
│   ├── ResourcesPage/
│   │   ├── index.ts
│   │   ├── ResourcesPage.tsx
│   │   ├── ResourcesPage.test.tsx
│   │   ├── ResourcesPage.module.css
│   │   └── types.ts
│   ├── RoadmapsPage/
│   │   ├── index.ts
│   │   ├── RoadmapsPage.tsx
│   │   ├── RoadmapsPage.test.tsx
│   │   ├── RoadmapsPage.module.css
│   │   └── types.ts
│   └── InterviewPrepPage/
│       ├── index.ts
│       ├── InterviewPrepPage.tsx
│       ├── InterviewPrepPage.test.tsx
│       ├── InterviewPrepPage.module.css
│       └── types.ts
├── hooks/
│   ├── useResources.ts
│   ├── useRoadmaps.ts
│   ├── useBlogs.ts
│   ├── useInterviewPrep.ts
│   └── index.ts
├── services/
│   ├── resourcesApi.ts
│   └── index.ts
├── store/
│   ├── resourcesSlice.ts
│   ├── resourcesSelectors.ts
│   └── index.ts
├── types/
│   ├── resource.types.ts
│   └── index.ts
└── index.ts
```

```
├── community/
│   ├── components/
│   │   ├── StudyGroupCard/
│   │   │   ├── index.ts
│   │   │   ├── StudyGroupCard.tsx
│   │   │   ├── StudyGroupCard.test.tsx
│   │   │   ├── StudyGroupCard.module.css
│   │   │   └── types.ts
│   │   ├── DiscussionThread/
│   │   │   ├── index.ts
│   │   │   ├── DiscussionThread.tsx
│   │   │   ├── DiscussionThread.test.tsx
│   │   │   ├── DiscussionThread.module.css
│   │   │   └── types.ts
│   │   ├── MentorshipCard/
│   │   │   ├── index.ts
│   │   │   ├── MentorshipCard.tsx
│   │   │   ├── MentorshipCard.test.tsx
│   │   │   ├── MentorshipCard.module.css
│   │   │   └── types.ts
│   │   ├── UserFollowButton/
│   │   │   ├── index.ts
│   │   │   ├── UserFollowButton.tsx
│   │   │   ├── UserFollowButton.test.tsx
│   │   │   ├── UserFollowButton.module.css
│   │   │   └── types.ts
│   │   └── CommunityFeed/
│   │       ├── index.ts
│   │       ├── CommunityFeed.tsx
│   │       ├── CommunityFeed.test.tsx
│   │       ├── CommunityFeed.module.css
│   │       └── types.ts
│   ├── pages/
│   │   ├── CommunityPage/
│   │   │   ├── index.ts
│   │   │   ├── CommunityPage.tsx
│   │   │   ├── CommunityPage.test.tsx
│   │   │   ├── CommunityPage.module.css
│   │   │   └── types.ts
│   │   ├── StudyGroupsPage/
│   │   │   ├── index.ts
│   │   │   ├── StudyGroupsPage.tsx
│   │   │   ├── StudyGroupsPage.test.tsx
│   │   │   ├── StudyGroupsPage.module.css
│   │   │   └── types.ts
│   │   └── MentorshipPage/
│   │       ├── index.ts
│   │       ├── MentorshipPage.tsx
│   │       ├── MentorshipPage.test.tsx
│   │       ├── MentorshipPage.module.css
│   │       └── types.ts
│   ├── hooks/
│   │   ├── useCommunity.ts
│   │   ├── useStudyGroups.ts
│   │   ├── useDiscussions.ts
│   │   └── useMentorship.ts
```

```
├── useFollowing.ts
├── index.ts
├── services/
│   ├── communityApi.ts
│   └── index.ts
├── store/
│   ├── communitySlice.ts
│   ├── communitySelectors.ts
│   └── index.ts
├── types/
│   ├── community.types.ts
│   └── index.ts
└── index.ts
├── settings/
│   ├── components/
│   │   ├── ProfileSettings/
│   │   │   ├── index.ts
│   │   │   ├── ProfileSettings.tsx
│   │   │   ├── ProfileSettings.test.tsx
│   │   │   ├── ProfileSettings.module.css
│   │   │   └── types.ts
│   │   ├── PrivacySettings/
│   │   │   ├── index.ts
│   │   │   ├── PrivacySettings.tsx
│   │   │   ├── PrivacySettings.test.tsx
│   │   │   ├── PrivacySettings.module.css
│   │   │   └── types.ts
│   │   ├── NotificationSettings/
│   │   │   ├── index.ts
│   │   │   ├── NotificationSettings.tsx
│   │   │   ├── NotificationSettings.test.tsx
│   │   │   ├── NotificationSettings.module.css
│   │   │   └── types.ts
│   │   ├── AccountSettings/
│   │   │   ├── index.ts
│   │   │   ├── AccountSettings.tsx
│   │   │   ├── AccountSettings.test.tsx
│   │   │   ├── AccountSettings.module.css
│   │   │   └── types.ts
│   │   └── ThemeSettings/
│   │       ├── index.ts
│   │       ├── ThemeSettings.tsx
│   │       ├── ThemeSettings.test.tsx
│   │       ├── ThemeSettings.module.css
│   │       └── types.ts
│   ├── pages/
│   │   └── SettingsPage/
│   │       ├── index.ts
│   │       ├── SettingsPage.tsx
│   │       ├── SettingsPage.test.tsx
│   │       ├── SettingsPage.module.css
│   │       └── types.ts
│   └── hooks/
│       ├── useSettings.ts
│       ├── useTheme.ts
│       └── useNotificationSettings.ts
```

```
└─ index.ts
└─ services/
  └─ settingsApi.ts
  └─ index.ts
└─ store/
  └─ settingsSlice.ts
  └─ settingsSelectors.ts
  └─ index.ts
└─ types/
  └─ settings.types.ts
  └─ index.ts
└─ index.ts
└─ shared/
  └─ components/
    └─ ErrorBoundary/
      └─ index.ts
      └─ ErrorBoundary.tsx
      └─ ErrorBoundary.test.tsx
      └─ ErrorBoundary.module.css
      └─ types.ts
    └─ LoadingSpinner/
      └─ index.ts
      └─ LoadingSpinner.tsx
      └─ LoadingSpinner.test.tsx
      └─ LoadingSpinner.module.css
      └─ types.ts
    └─ SEOWrapper/
      └─ index.ts
      └─ SEOWrapper.tsx
      └─ SEOWrapper.test.tsx
      └─ types.ts
    └─ LazyLoadWrapper/
      └─ index.ts
      └─ LazyLoadWrapper.tsx
      └─ LazyLoadWrapper.test.tsx
      └─ types.ts
  └─ hooks/
    └─ useLocalStorage.ts
    └─ useSessionStorage.ts
    └─ useDebounce.ts
    └─ useThrottle.ts
    └─ useIntersectionObserver.ts
    └─ useKeyPress.ts
    └─ useClickOutside.ts
    └─ useWindowSize.ts
    └─ useMediaQuery.ts
    └─ usePrevious.ts
    └─ useToggle.ts
    └─ useInterval.ts
    └─ useTimeout.ts
    └─ useCopyToClipboard.ts
    └─ useOnlineStatus.ts
    └─ index.ts
  └─ utils/
    └─ api.ts
    └─ auth.ts
```



```

├── validation.ts
├── formatters.ts
├── constants.ts
├── helpers.ts
├── storage.ts
├── analytics.ts
├── seo.ts
├── performance.ts
├── index.ts
├── index.ts
├── index.ts
├── pages/
├──   HomePage/
├──     ├── index.ts
├──     ├── HomePage.tsx
├──     ├── HomePage.test.tsx
├──     ├── HomePage.module.css
├──     └── types.ts
├──   LandingPage/
├──     ├── index.ts
├──     ├── LandingPage.tsx
├──     ├── LandingPage.test.tsx
├──     ├── LandingPage.module.css
├──     └── types.ts
├──   NotFoundPage/
├──     ├── index.ts
├──     ├── NotFoundPage.tsx
├──     ├── NotFoundPage.test.tsx
├──     ├── NotFoundPage.module.css
├──     └── types.ts
├──   index.ts
├── store/
├──   ├── index.ts
├──   ├── rootReducer.ts
├──   ├── store.ts
├──   ├── middleware.ts
├──   ├── types.ts
├──   └── selectors.ts
├── services/
├──   ├── api/
├──     ├── index.ts
├──     ├── client.ts
├──     ├── endpoints.ts
├──     ├── interceptors.ts
├──     ├── types.ts
├──     └── config.ts
├──   ├── websocket/
├──     ├── index.ts
├──     ├── websocketClient.ts
├──     ├── websocketHandlers.ts
├──     └── types.ts
├──   ├── storage/
├──     ├── index.ts
├──     ├── localStorage.ts
├──     ├── sessionStorage.ts
├──     └── indexedDB.ts

```

- analytics/
 - index.ts
 - googleAnalytics.ts
 - mixpanel.ts
 - customEvents.ts
- push-notifications/
 - index.ts
 - fcm.ts
 - serviceWorker.ts
- index.ts
- types/
 - global/
 - index.ts
 - api.types.ts
 - common.types.ts
 - user.types.ts
 - navigation.types.ts
 - theme.types.ts
 - enums/
 - index.ts
 - UserRole.enum.ts
 - ProblemDifficulty.enum.ts
 - SubmissionStatus.enum.ts
 - ContestType.enum.ts
 - Language.enum.ts
 - BadgeType.enum.ts
 - NotificationType.enum.ts
 - index.ts
- hooks/
 - api/
 - index.ts
 - useApi.ts
 - useQuery.ts
 - useMutation.ts
 - useInfiniteQuery.ts
 - auth/
 - index.ts
 - useAuthRedirect.ts
 - usePermissions.ts
 - useProtectedRoute.ts
 - ui/
 - index.ts
 - useModal.ts
 - useToast.ts
 - usePagination.ts
 - useSearch.ts
 - useSort.ts
 - useFilters.ts
 - websocket/
 - index.ts
 - useWebSocket.ts
 - useLiveUpdates.ts
 - useNotifications.ts
 - index.ts
- contexts/
 - ThemeContext/

- index.ts
 - ThemeContext.tsx
 - ThemeProvider.tsx
 - ThemeContext.test.tsx
 - types.ts
- AuthContext/
 - index.ts
 - AuthContext.tsx
 - AuthProvider.tsx
 - AuthContext.test.tsx
 - types.ts
- WebSocketContext/
 - index.ts
 - WebSocketContext.tsx
 - WebSocketProvider.tsx
 - types.ts
- index.ts
- routes/
 - index.ts
 - AppRoutes.tsx
 - ProtectedRoute.tsx
 - PublicRoute.tsx
 - LazyRoutes.tsx
 - routeConfig.ts
 - types.ts
- styles/
 - globals.css
 - variables.css
 - reset.css
 - typography.css
 - layout.css
 - components.css
 - utilities.css
 - animations.css
 - responsive.css
 - themes/
 - light.css
 - dark.css
 - index.css
 - vendor/
 - monaco-editor.css
 - chart.css
 - prism.css
- config/
 - index.ts
 - env.ts
 - api.config.ts
 - auth.config.ts
 - theme.config.ts
 - analytics.config.ts
 - websocket.config.ts
 - storage.config.ts
 - feature-flags.config.ts
- constants/
 - index.ts
 - api.constants.ts

```
|
|
|   |— ui.constants.ts
|   |— auth.constants.ts
|   |— routing.constants.ts
|   |— validation.constants.ts
|   |— storage.constants.ts
|   |— analytics.constants.ts
|— utils/
|   |— test/
|   |   |— index.ts
|   |   |— testUtils.tsx
|   |   |— mockData.ts
|   |   |— setup.ts
|   |   |— renderWithProviders.tsx
|   |— index.ts
|— __tests__/
|   |— setup.ts
|   |— mocks/
|   |   |— handlers.ts
|   |   |— server.ts
|   |   |— localStorage.mock.ts
|   |   |— websocket.mock.ts
|   |   |— api.mock.ts
|   |— fixtures/
|   |   |— user.fixtures.ts
|   |   |— problem.fixtures.ts
|   |   |— contest.fixtures.ts
|   |   |— analytics.fixtures.ts
|   |— integration/
|   |   |— auth.test.tsx
|   |   |— problem-solving.test.tsx
|   |   |— contest.test.tsx
|   |   |— dashboard.test.tsx
|— cypress/
|   |— e2e/
|   |   |— auth.cy.ts
|   |   |— onboarding.cy.ts
|   |   |— problem-solving.cy.ts
|   |   |— contests.cy.ts
|   |   |— leaderboard.cy.ts
|   |   |— user-journey.cy.ts
|   |— fixtures/
|   |   |— users.json
|   |   |— problems.json
|   |   |— contests.json
|   |— support/
|   |   |— commands.ts
|   |   |— e2e.ts
|   |   |— component.ts
|   |— cypress.config.ts
|— docs/
|   |— SETUP.md
|   |— DEVELOPMENT.md
|   |— DEPLOYMENT.md
|   |— TESTING.md
|   |— CONTRIBUTING.md
|   |— API.md
```

```

├── ARCHITECTURE.md
├── STYLE_GUIDE.md
├── scripts/
│   ├── build.sh
│   ├── test.sh
│   ├── deploy.sh
│   ├── generate-types.js
│   ├── analyze-bundle.js
│   └── setup-env.js
├── .env
├── .env.development
├── .env.production
├── .env.test
├── .gitignore
├── .eslintrc.js
├── .prettierrc
├── .dockerignore
├── Dockerfile
├── docker-compose.yml
├── nginx.conf
├── package.json
├── package-lock.json
├── tsconfig.json
├── tsconfig.paths.json
├── tailwind.config.js
├── postcss.config.js
├── vite.config.ts
├── vitest.config.ts
├── jest.config.js
├── webpack.config.js (if not using Vite)
├── README.md
├── LICENSE
└── CHANGELOG.md

```

Database Schema (PostgreSQL/MySQL)

```

-- User Management Tables
CREATE TABLE users (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    email VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    username VARCHAR(100) UNIQUE NOT NULL,
    is_verified BOOLEAN DEFAULT FALSE,
    is_active BOOLEAN DEFAULT TRUE,
    role VARCHAR(50) DEFAULT 'USER',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE user_profiles (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    full_name VARCHAR(255) NOT NULL,
    prn VARCHAR(100) UNIQUE NOT NULL, -- College ID
    college_name VARCHAR(255) NOT NULL,

```

```

    year_of_passing INTEGER NOT NULL,
    profile_picture_url TEXT,
    resume_url TEXT,
    bio TEXT,
    location VARCHAR(255),
    github_url TEXT,
    linkedin_url TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE competitive_programming_profiles (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    leetcode_username VARCHAR(100),
    leetcode_profile_url TEXT,
    codeforces_username VARCHAR(100),
    codeforces_profile_url TEXT,
    codechef_username VARCHAR(100),
    codechef_profile_url TEXT,
    geeksforgeeks_username VARCHAR(100),
    geeksforgeeks_profile_url TEXT,
    programming_level VARCHAR(50) DEFAULT 'BEGINNER', -- BEGINNER, INTERMEDIATE, ADVANCED
    daily_practice_target INTEGER DEFAULT 1,
    preferred_languages TEXT[], -- Array of programming languages
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Problem & Sheet Management
CREATE TABLE problems (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    title VARCHAR(255) NOT NULL,
    description TEXT NOT NULL,
    difficulty VARCHAR(50) NOT NULL, -- EASY, MEDIUM, HARD
    topics TEXT[] NOT NULL, -- Array of topics
    companies TEXT[], -- Array of company names
    acceptance_rate DECIMAL(5,2),
    time_limit INTEGER DEFAULT 1000, -- in milliseconds
    memory_limit INTEGER DEFAULT 256, -- in MB
    editorial_url TEXT,
    video_solution_url TEXT,
    created_by UUID REFERENCES users(id),
    is_premium BOOLEAN DEFAULT FALSE,
    is_active BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE problem_tags (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    problem_id UUID NOT NULL REFERENCES problems(id) ON DELETE CASCADE,
    tag_name VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```
CREATE TABLE test_cases (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  problem_id UUID NOT NULL REFERENCES problems(id) ON DELETE CASCADE,
  input TEXT NOT NULL,
  expected_output TEXT NOT NULL,
  is_hidden BOOLEAN DEFAULT FALSE, -- For contest problems
  weight INTEGER DEFAULT 1, -- For partial scoring
  explanation TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TABLE sheets (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  name VARCHAR(255) NOT NULL,
  description TEXT,
  type VARCHAR(50) NOT NULL, -- STRIVER, BLIND75, CUSTOM, etc.
  difficulty_level VARCHAR(50),
  estimated_days INTEGER,
  total_problems INTEGER DEFAULT 0,
  created_by UUID REFERENCES users(id),
  is_public BOOLEAN DEFAULT TRUE,
  banner_image_url TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TABLE sheet_problems (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  sheet_id UUID NOT NULL REFERENCES sheets(id) ON DELETE CASCADE,
  problem_id UUID NOT NULL REFERENCES problems(id) ON DELETE CASCADE,
  order_index INTEGER NOT NULL,
  category VARCHAR(100), -- Arrays, Trees, etc.
  is_mandatory BOOLEAN DEFAULT TRUE,
  points INTEGER DEFAULT 1,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

-- Submission & Code Execution

```
CREATE TABLE languages (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  name VARCHAR(50) NOT NULL UNIQUE,
  version VARCHAR(20),
  file_extension VARCHAR(10),
  compile_command TEXT,
  execute_command TEXT,
  is_active BOOLEAN DEFAULT TRUE
);
```

```
CREATE TABLE submissions (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID NOT NULL REFERENCES users(id),
  problem_id UUID NOT NULL REFERENCES problems(id),
  language_id UUID NOT NULL REFERENCES languages(id),
  code TEXT NOT NULL,
  status VARCHAR(50) NOT NULL, -- PENDING, ACCEPTED, WRONG_ANSWER, TIME_LIMIT_EXCEEDED,
  execution_time INTEGER, -- in milliseconds
);
```

```

memory_used INTEGER, -- in KB
test_cases_passed INTEGER DEFAULT 0,
total_test_cases INTEGER DEFAULT 0,
error_message TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE user_problem_stats (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id),
    problem_id UUID NOT NULL REFERENCES problems(id),
    is_solved BOOLEAN DEFAULT FALSE,
    attempts_count INTEGER DEFAULT 0,
    best_submission_id UUID REFERENCES submissions(id),
    first_solved_at TIMESTAMP,
    last_attempt_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    time_spent_seconds INTEGER DEFAULT 0,
    UNIQUE(user_id, problem_id)
);

-- Contest System
CREATE TABLE contests (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    title VARCHAR(255) NOT NULL,
    description TEXT,
    type VARCHAR(50) NOT NULL, -- WEEKLY, MONTHLY, SPECIAL
    start_time TIMESTAMP NOT NULL,
    end_time TIMESTAMP NOT NULL,
    duration_minutes INTEGER NOT NULL,
    max_participants INTEGER,
    registration_deadline TIMESTAMP,
    isRated BOOLEAN DEFAULT TRUE,
    is_public BOOLEAN DEFAULT TRUE,
    banner_image_url TEXT,
    rules TEXT,
    prizes TEXT,
    created_by UUID REFERENCES users(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE contest_problems (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    contest_id UUID NOT NULL REFERENCES contests(id) ON DELETE CASCADE,
    problem_id UUID NOT NULL REFERENCES problems(id),
    order_index INTEGER NOT NULL,
    points INTEGER DEFAULT 100,
    penalty_minutes INTEGER DEFAULT 10,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE contest_participations (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    contest_id UUID NOT NULL REFERENCES contests(id),
    user_id UUID NOT NULL REFERENCES users(id),
    registration_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

```



```

    start_time TIMESTAMP,
    end_time TIMESTAMP,
    total_score INTEGER DEFAULT 0,
    rank_position INTEGER,
    problems_solved INTEGER DEFAULT 0,
    penalty_time INTEGER DEFAULT 0,
    UNIQUE(contest_id, user_id)
);

CREATE TABLE contest_submissions (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    participation_id UUID NOT NULL REFERENCES contest_participations(id),
    problem_id UUID NOT NULL REFERENCES problems(id),
    submission_id UUID NOT NULL REFERENCES submissions(id),
    points_earned INTEGER DEFAULT 0,
    penalty_time INTEGER DEFAULT 0,
    submission_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    is_accepted BOOLEAN DEFAULT FALSE
);

-- Analytics & Gamification
CREATE TABLE user_daily_activities (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id),
    activity_date DATE NOT NULL,
    problems_solved INTEGER DEFAULT 0,
    time_spent_seconds INTEGER DEFAULT 0,
    submissions_count INTEGER DEFAULT 0,
    contests_participated INTEGER DEFAULT 0,
    study_streak_days INTEGER DEFAULT 0,
    xp_earned INTEGER DEFAULT 0,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE(user_id, activity_date)
);

CREATE TABLE user_topic_performance (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id),
    topic_name VARCHAR(100) NOT NULL,
    problems_solved INTEGER DEFAULT 0,
    problems_attempted INTEGER DEFAULT 0,
    success_rate DECIMAL(5,2) DEFAULT 0.00,
    average_time_seconds INTEGER DEFAULT 0,
    difficulty_breakdown JSONB, -- {easy: 5, medium: 3, hard: 1}
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE(user_id, topic_name)
);

CREATE TABLE badges (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(100) NOT NULL UNIQUE,
    description TEXT,
    icon_url TEXT,
    type VARCHAR(50) NOT NULL, -- MILESTONE, STREAK, CONTEST, SPECIAL
    criteria JSONB NOT NULL, -- Conditions to earn badge
    rarity VARCHAR(50) DEFAULT 'COMMON', -- COMMON, RARE, EPIC, LEGENDARY

```

```

        is_active BOOLEAN DEFAULT TRUE,
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    );

CREATE TABLE user_badges (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id),
    badge_id UUID NOT NULL REFERENCES badges(id),
    earned_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    progress JSONB, -- For progressive badges
    UNIQUE(user_id, badge_id)
);

CREATE TABLE user_levels (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) UNIQUE,
    current_level INTEGER DEFAULT 1,
    current_xp INTEGER DEFAULT 0,
    total_xp INTEGER DEFAULT 0,
    next_level_xp INTEGER DEFAULT 100,
    level_title VARCHAR(100) DEFAULT 'Beginner',
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Leaderboards
CREATE TABLE global_rankings (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id),
    rank_position INTEGER NOT NULL,
    total_score INTEGER DEFAULT 0,
    problems_solved INTEGER DEFAULT 0,
    contest_rating INTEGER DEFAULT 1200,
    contribution_points INTEGER DEFAULT 0,
    ranking_type VARCHAR(50) DEFAULT 'OVERALL', -- OVERALL, MONTHLY, WEEKLY
    ranking_period VARCHAR(50), -- YYYY-MM for monthly, YYYY-WW for weekly
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE(user_id, ranking_type, ranking_period)
);

CREATE TABLE college_rankings (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id),
    college_name VARCHAR(255) NOT NULL,
    rank_position INTEGER NOT NULL,
    total_score INTEGER DEFAULT 0,
    problems_solved INTEGER DEFAULT 0,
    ranking_period VARCHAR(50),
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Community Features
CREATE TABLE study_groups (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(255) NOT NULL,
    description TEXT,
    created_by UUID NOT NULL REFERENCES users(id),

```

```

    max_members INTEGER DEFAULT 50,
    is_public BOOLEAN DEFAULT TRUE,
    focus_topics TEXT[], -- Array of topics
    skill_level VARCHAR(50), -- BEGINNER, INTERMEDIATE, ADVANCED, MIXED
    meeting_schedule VARCHAR(255),
    invite_code VARCHAR(20) UNIQUE,
    banner_image_url TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```
);
```

```

CREATE TABLE study_group_members (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    group_id UUID NOT NULL REFERENCES study_groups(id) ON DELETE CASCADE,
    user_id UUID NOT NULL REFERENCES users(id),
    role VARCHAR(50) DEFAULT 'MEMBER', -- ADMIN, MODERATOR, MEMBER
    joined_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    is_active BOOLEAN DEFAULT TRUE,
    UNIQUE(group_id, user_id)
);

```

```
);
```

```

CREATE TABLE discussions (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    title VARCHAR(255) NOT NULL,
    content TEXT NOT NULL,
    problem_id UUID REFERENCES problems(id),
    created_by UUID NOT NULL REFERENCES users(id),
    category VARCHAR(100), -- QUESTION, SOLUTION, GENERAL
    tags TEXT[],
    upvotes_count INTEGER DEFAULT 0,
    views_count INTEGER DEFAULT 0,
    replies_count INTEGER DEFAULT 0,
    is_pinned BOOLEAN DEFAULT FALSE,
    is_locked BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```
);
```

```

CREATE TABLE discussion_replies (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    discussion_id UUID NOT NULL REFERENCES discussions(id) ON DELETE CASCADE,
    content TEXT NOT NULL,
    created_by UUID NOT NULL REFERENCES users(id),
    parent_reply_id UUID REFERENCES discussion_replies(id),
    upvotes_count INTEGER DEFAULT 0,
    is_solution BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at

```

[^1][^10][^11][^12][^13][^14][^15][^16][^17][^18][^19][^2][^20

<div style="text-align: center">✱✱</div>

[^1]: <https://www.jetir.org/papers/JETIR2502735.pdf>

[^2]: <https://www.irjweb.com/LEET%20CODE%20CLONE.pdf>

[^3]: <https://www.geeksforgeeks.org/blogs/geeksforgeeks-practice-best-online-coding-platf>

[^4]: <https://arxiv.org/pdf/2311.10244.pdf>

[^5]: https://www.reddit.com/r/learnprogramming/comments/16ye9x1/how_to_make_a_leetcode_c
[^6]: <https://fullscale.io/blog/top-5-tech-stacks/>
[^7]: <https://www.geeksforgeeks.org/dsa/competitive-programming-a-complete-guide/>
[^8]: https://www.linkedin.com/posts/prasanna-h-28b07b27b_i-built-a-leetcode-clone-with-t
[^9]: <https://www.heap.io/topics/what-is-a-tech-stack>
[^10]: https://www.reddit.com/r/ProgrammingLanguages/comments/19m4cw/i_want_to_design_and
[^11]: <https://www.enacton.com/leetcode-clone/>
[^12]: <https://coderpad.io>
[^13]: <https://www.youtube.com/watch?v=GnodscC2p-A>
[^14]: <https://www.codechef.com>
[^15]: https://www.epj-conferences.org/articles/epjconf/pdf/2014/05/epjconf_icas2013_0001
[^16]: <https://www.geeksforgeeks.org/dbms/database-schemas/>
[^17]: <https://www.youtube.com/watch?v=T0yD-5QgpuE>
[^18]: <https://tianpan.co/notes/243-designing-online-judge-or-leetcode>
[^19]: <https://www.ijcrt.org/papers/IJCRTAF02027.pdf>
[^20]: <https://www.geeksforgeeks.org/java/execution-engine-in-java/>
[^21]: <https://blog.huli.tw/2020/03/23/en/build-your-own-online-judge-system/>
[^22]: <https://www.designgurus.io/blog/advanced-data-structure-patterns-for-competitive-c>
[^23]: <https://dev.to/acoh3n/lets-build-a-code-execution-engine-4kgi>
[^24]: <https://jyotinder.substack.com/p/build-a-remote-code-execution-engine>
[^25]: <https://blog.devgenius.io/case-study-remote-code-execution-engine-system-63aa43344>
[^26]: <https://www.youtube.com/watch?v=SD4KgwdjmdI>
[^27]: <https://www.robinwieruch.de/react-folder-structure/>
[^28]: <https://pwskills.com/blog/architecture-of-spring-boot-examples-pattern-layered-cor>
[^29]: <https://www.youtube.com/watch?v=xyxrB2Aa7KE>
[^30]: <https://stackademic.com/blog/react-typescript-folder-structure-to-follow-ae614e786>
[^31]: <https://www.geeksforgeeks.org/springboot/spring-boot-architecture/>
[^32]: <https://dev.to/noruwa/folder-structure-for-modern-web-applications-4d11>
[^33]: <https://blog.bitsrc.io/how-to-set-up-a-react-project-with-typescript-842307b9504d>
[^34]: <https://dzone.com/articles/intro-to-distributed-system-patterns-in-spring-boot>
[^35]: <https://nextjs.org/docs/app/getting-started/project-structure>
[^36]: https://www.reddit.com/r/reactjs/comments/151q4d2/what_is_your_preferred_folder_st
[^37]: <https://www.baeldung.com/spring-boot-clean-architecture>
[^38]: <https://www.netguru.com/blog/react-project-structure>
[^39]: <https://scrimba.com/articles/react-project-structure/>