**📝 TÀI LIỆU HƯỚNG DẪN DỰ ÁN TOEIC WRITING TEST**

**🚀 THÔNG TIN DỰ ÁN**

| **Trường** | **Giá trị** |
| --- | --- |
| **Tên dự án** | **TOEIC Writing Test - Ứng dụng Luyện Thi Viết TOEIC** |
| **Phiên bản** | 1.0.0 |
| **Ngày tạo** | October 2025 |
| **Công nghệ** | Angular 20.3.0 (Standalone Components) |
| **Tác giả** | Development Team |
| **Mục đích** | Cung cấp nền tảng luyện thi TOEIC Writing trực tuyến với giao diện thân thiện và chức năng đầy đủ |

**📜 MỤC LỤC**

1. [TỔNG QUAN DỰ ÁN](https://www.google.com/search?q=%231-t%E1%BB%95ng-quan-d%E1%BB%B1-%C3%A1n)
2. [KIẾN TRÚC HỆ THỐNG](https://www.google.com/search?q=%232-ki%E1%BA%BFn-tr%C3%BAc-h%E1%BB%87-th%E1%BB%91ng)
3. [CÁC TÍNH NĂNG CHÍNH](https://www.google.com/search?q=%233-c%C3%A1c-t%C3%ADnh-n%C4%83ng-ch%C3%ADnh)
4. [MÔ HÌNH DỮ LIỆU](https://www.google.com/search?q=%234-m%C3%B4-h%C3%ACnh-d%E1%BB%AF-li%E1%BB%87u)
5. [GIAO DIỆN NGƯỜI DÙNG](https://www.google.com/search?q=%235-giao-di%E1%BB%87n-ng%C6%B0%E1%BB%9Di-d%C3%B9ng)
6. [LUỒNG HOẠT ĐỘNG](https://www.google.com/search?q=%236-lu%E1%BB%93ng-ho%E1%BA%A1t-%C4%91%E1%BB%99ng)
7. [HƯỚNG DẪN CÀI ĐẶT](https://www.google.com/search?q=%237-h%C6%B0%E1%BB%9Bng-d%E1%BA%ABn-c%C3%A0i-%C4%91%E1%BA%B7t)
8. [HƯỚNG DẪN PHÁT TRIỂN](https://www.google.com/search?q=%238-h%C6%B0%E1%BB%9Bng-d%E1%BA%ABn-ph%C3%A1t-tri%E1%BB%83n)
   * [PHỤ LỤC](https://www.google.com/search?q=%23ph%E1%BB%A5-l%E1%BB%A5c)

**1. TỔNG QUAN DỰ ÁN**

**1.1. Giới Thiệu**

**TOEIC Writing Test** là một ứng dụng web được xây dựng bằng **Angular 20.3.0**, sử dụng kiến trúc **Standalone Components** hiện đại. Ứng dụng mô phỏng bài thi TOEIC Writing thực tế, giúp người học luyện tập và cải thiện kỹ năng viết tiếng Anh.

**1.2. Mục Tiêu**

* ✅ Cung cấp trải nghiệm thi thử TOEIC Writing **chân thực**
* ✅ Hỗ trợ **3 dạng câu hỏi**: Write a Sentence, Respond to Email, Essay
* ✅ Tính **thời gian làm bài tự động**
* ✅ **Lưu trữ và xem lại kết quả**
* ✅ Giao diện thân thiện, **responsive**
* ✅ Hỗ trợ **đếm từ tự động**

**1.3. Phạm Vi**

Dự án bao gồm các thành phần chính:

* Trang chủ (tự động chuyển hướng đến trang thi)
* **Trang làm bài thi** (Test Component)
* **Trang xem kết quả** (Result Component)
* Service quản lý trạng thái bài thi (TestService)
* Models định nghĩa cấu trúc dữ liệu

**2. KIẾN TRÚC HỆ THỐNG**

**2.1. Công Nghệ Sử Dụng**

| **Công nghệ** | **Phiên bản** | **Mục đích** |
| --- | --- | --- |
| **Angular** | 20.3.0 | Framework chính (Frontend) |
| **TypeScript** | 5.9.2 | Ngôn ngữ lập trình |
| **SCSS** | Latest | Styling |
| **Angular Signals** | Built-in | Quản lý state (State Management) |
| **Express.js** | 4.21.2 | Server-side rendering (SSR) |
| **RxJS** | 7.8.0 | Reactive programming |

**2.2. Kiến Trúc Components**

Hệ thống tuân thủ mô hình **Component-Service** và sử dụng **Angular Signals** để quản lý trạng thái tập trung.

**3. CÁC TÍNH NĂNG CHÍNH**

**3.1. Chức Năng Làm Bài Thi**

**3.1.1. Quản Lý Thời Gian**

* **Timer tự động**: Đếm ngược từ **60 phút** (3600 giây).
* **Hiển thị thời gian**: Format **MM:SS** (ví dụ: 59:45).
* **Cảnh báo**: Thay đổi màu (đỏ) khi còn **< 5 phút**.
* **Tự động submit**: Khi hết thời gian, bài thi sẽ tự động được nộp.

**3.1.2. Điều Hướng Câu Hỏi**

* **Navigation Buttons**: Nút Next và Previous.
* **Sidebar Navigation**: Click vào số thứ tự câu hỏi.
* **Keyboard Shortcuts**:
  + Ctrl + → : **Next** Question
  + Ctrl + ← : **Previous** Question

**3.1.3. Tự Động Lưu (Auto Save)**

* Lưu câu trả lời khi người dùng gõ (**debounce 500ms**).
* Lưu vào **LocalStorage** để đảm bảo dữ liệu không bị mất khi refresh trang.

**3.1.4. Đếm Từ**

* Đếm số từ **real-time** và hiển thị cho từng câu trả lời.
* Cảnh báo nếu câu trả lời không đạt **số từ tối thiểu** yêu cầu.

**3.2. Chức Năng Xem Kết Quả**

**3.2.1. Thống Kê Tổng Quan**

* Tổng số câu hỏi: **8**
* Số câu đã trả lời và Tỷ lệ hoàn thành (%).
* **Điểm ước tính (0-200)**.
* Thời gian hoàn thành và **Tổng số từ đã viết**.

**3.2.2. Chi Tiết Từng Dạng Câu**

| **Dạng câu hỏi** | **Số câu** | **Yêu cầu** | **Thời gian gợi ý** | **Số từ yêu cầu** |
| --- | --- | --- | --- | --- |
| **Write a Sentence** | 5 (Q1-5) | Viết 1 câu từ từ cho sẵn | 8 phút | 10-15 từ/câu |
| **Respond to Email** | 2 (Q6-7) | Trả lời email | 10 phút/câu | ≥ 25 từ/câu |
| **Essay** | 1 (Q8) | Viết bài luận | 30 phút | **≥ 300 từ** |

**3.2.3. Tabs Hiển Thị**

Kết quả được tổ chức trong 4 tab: **Question-8 Tab**, **Questions 1-5 Tab**, **Questions 6-7 Tab**, và **Summary Tab** (Tổng hợp nhận xét/Điểm số).

**4. MÔ HÌNH DỮ LIỆU**

**4.1. QuestionType Enum**

TypeScript

export enum QuestionType {

  WRITE\_SENTENCE = 'WRITE\_SENTENCE',    // Câu 1-5

  RESPOND\_EMAIL = 'RESPOND\_EMAIL',      // Câu 6-7

  ESSAY = 'ESSAY'                       // Câu 8

}

**4.2. Question Interface**

TypeScript

export interface Question {

  id: number;                    // ID câu hỏi (1-8)

  type: QuestionType;            // Loại câu hỏi

  title: string;                 // Tiêu đề câu hỏi

  description: string;           // Mô tả yêu cầu

  imageUrl?: string;             // URL hình ảnh (nếu có)

  words?: string[];              // Từ cho sẵn (WRITE\_SENTENCE)

  emailContent?: string;         // Nội dung email (RESPOND\_EMAIL)

  minWords: number;              // Số từ tối thiểu

  suggestedTime: number;         // Thời gian gợi ý (phút)

}

**4.3. Answer Interface**

TypeScript

export interface Answer {

  questionId: number;            // ID câu hỏi

  content: string;               // Nội dung câu trả lời

  wordCount: number;             // Số từ đã viết

  timeSpent: number;             // Thời gian làm bài (giây)

  isCompleted: boolean;          // Đã hoàn thành chưa

}

**4.4. TestSession Interface**

TypeScript

export interface TestSession {

  sessionId: string;             // ID phiên thi

  startTime: Date;               // Thời gian bắt đầu

  endTime?: Date;                // Thời gian kết thúc

  totalTime: number;             // Tổng thời gian (3600s)

  remainingTime: number;         // Thời gian còn lại

  currentQuestionId: number;     // Câu hỏi hiện tại

  answers: Map<number, Answer>;  // Map câu trả lời

  isSubmitted: boolean;          // Đã submit chưa

  estimatedScore?: number;       // Điểm ước tính

}

**5. GIAO DIỆN NGƯỜI DÙNG**

**5.1. Màu Sắc Chủ Đạo (SCSS Variables)**

| **Biến SCSS** | **Mã màu** | **Ý nghĩa** |
| --- | --- | --- |
| $primary-green-start | #56ab2f | Màu chủ đạo (Bắt đầu Gradient) |
| $bg-white | #ffffff | Nền chính |
| $text-dark | #2c3e50 | Văn bản chính |
| $answered | #4caf50 | Trạng thái: Đã trả lời (Xanh lá) |
| $current | #2196f3 | Trạng thái: Câu hiện tại (Xanh dương) |
| $danger | #f44336 | Trạng thái: Nguy hiểm/Cảnh báo đỏ |

**5.2. Layout Test Page**

Thiết kế hai cột: **Main Content Area** (75%) và **Question Sidebar** (25%).

**5.3. Layout Result Page**

Layout tập trung vào **Statistics** (Thống kê) ở trên và **Tabs** (Chi tiết câu hỏi) ở dưới.

**6. LUỒNG HOẠT ĐỘNG**

**6.1. User Flow Diagram**

Sơ đồ thể hiện luồng chính của người dùng:

Đoạn mã

graph TD

A[START /] --> B{Auto Redirect};

B --> C[TEST PAGE /test];

C --> D[Initialize Test];

D --> E{User Actions: Type, Next, Prev, Jump};

E -- Auto Save (500ms) --> C;

E -- Submit / Timer = 0 --> F{Save Session};

F --> G[RESULT PAGE /result];

G --> H[Load Results & Render Tabs];

H -- Switch Tabs --> H;

H -- New Test --> D;

H --> I[END];

**6.2. State Management Flow**

Sử dụng TestService là **Singleton** để quản lý trạng thái tập trung với **Angular Signals**:

┌─────────────────────────────────────────────┐

│           \*\*TestService (Singleton)\*\*           │

├─────────────────────────────────────────────┤

│   \*\*Signals (Reactive State):\*\*              │

│   ├─ questions: WritableSignal<Question[]>   │

│   ├─ answers: WritableSignal<Map>            │

│   └─ remainingTime: WritableSignal<number>   │

│                                             │

│   \*\*Methods:\*\* initializeTest(), saveAnswer(), submitTest() │

└─────────────────┬─────────────────┘

                  │                          │

                  ▼                          ▼

         ┌────────────────┐          ┌─────────────────┐

         │ \*\*TestComponent\*\*│          │ \*\*ResultComponent\*\* │

         │ - Subscribe & Update UI│          │ - Subscribe & Display   │

         └────────────────┘          └─────────────────┘

**6.3. Timer Flow**

Sử dụng setInterval hoặc RxJS Timer để đếm ngược mỗi giây và kiểm tra điều kiện kết thúc:

Test Start -> Set Total Time (3600s) -> Start Interval (1s) -> Remaining Time -= 1 -> \*\*[IF Time > 0]\*\* Continue Loop / \*\*[ELSE]\*\* Auto Submit Test

**7. HƯỚNG DẪN CÀI ĐẶT**

**7.1. Yêu Cầu Hệ Thống**

* **Node.js**: **>= 18.19.0**
* **npm**: **>= 10.2.3**
* **Browser**: Chrome, Firefox, Safari, Edge (latest)
* **OS**: Windows, macOS, Linux

**7.2. Các Bước Cài Đặt**

**Bước 1: Clone Project**

Bash

git clone <repository-url>

cd writing\_test

**Bước 2: Install Dependencies**

Bash

npm install

**Bước 3: Run Development Server**

Bash

npm start

# hoặc

ng serve

**Bước 4: Mở Trình Duyệt**

Truy cập http://localhost:4200

**7.3. Build Production**

| **Mục đích** | **Lệnh** |
| --- | --- |
| **Build for production** | npm run build |
| **Build with SSR** | npm run build:ssr |
| **Serve SSR** | npm run serve:ssr |

**8. HƯỚNG DẪN PHÁT TRIỂN**

**8.1. Coding Standards**

* **TypeScript**: **strict mode**, biến camelCase, class/interface PascalCase, sử dụng arrow functions.
* **SCSS**: Sử dụng **BEM naming convention**, tổ chức theo components, sử dụng **variables**.
* **Responsive Design**: Ưu tiên phong cách **Mobile-first**.

**8.2. Component Development**

Sử dụng **Standalone Components**:

Bash

ng generate component components/my-component --standalone

**8.3. Service Development**

Service sử dụng **Angular Signals** cho quản lý state nội bộ.

TypeScript

@Injectable({

  providedIn: 'root'

})

export class MyService {

  private \_data = signal<any>(null);   // Private WritableSignal

  readonly data = this.\_data.asReadonly(); // Public ReadonlySignal

  constructor() {}

  updateData(value: any) {

    this.\_data.set(value);

  }

}

**8.4. Routing Configuration**

Sử dụng **Lazy Loading** cho components:

TypeScript

export const routes: Routes = [

  { path: '', redirectTo: '/test', pathMatch: 'full' },

  {

    path: 'test',

    loadComponent: () =>

      import('./components/test/test.component').then(m => m.TestComponent)

  },

  {

    path: 'result',

    loadComponent: () =>

      import('./components/result/result.component').then(m => m.ResultComponent)

  }

];

**8.5. Testing Guidelines**

| **Loại Test** | **Lệnh** | **Ghi chú** |
| --- | --- | --- |
| **Unit Tests** | ng test | Test logic của components, services, và pipes. |
| **Test Coverage** | ng test --code-coverage | Báo cáo tỷ lệ bao phủ code. |
| **E2E Tests** | ng e2e | Test luồng người dùng cuối. |

**8.6. Git Workflow**

Bash

git checkout -b feature/new-feature # Feature branch

git add .

git commit -m "feat: add new feature" # Commit theo Conventional Commits

git push origin feature/new-feature

# Tạo Pull Request, Review & Merge

**8.7. Best Practices**

* **Performance**: OnPush change detection, Lazy load routes, Optimize images.
* **Accessibility**: Semantic HTML, ARIA labels, Color contrast ratio **>= 4.5:1**.
* **Security**: **Sanitize user inputs**, dùng XSS protection có sẵn của Angular.
* **Code Quality**: Theo dõi Angular style guide, Code reviews thường xuyên.

**🏷️ PHỤ LỤC**

**A. Shortcuts Reference**

| **Shortcut** | **Action** |
| --- | --- |
| **Ctrl + →** | Next Question |
| **Ctrl + ←** | Previous Question |
| **Ctrl + Enter** | Submit Test |

**B. Browser Support**

* **Chrome**: >= 90
* **Firefox**: >= 88
* **Safari**: >= 14
* **Edge**: >= 90

**C. API Reference**

Các phương thức chính của **TestService**:

* initializeTest(): void
* goToNext(): void, goToPrevious(): void, goToQuestion(id: number): void
* saveAnswer(questionId: number, content: string): void
* submitTest(): void
* calculateScore(): number

**D. File Naming Conventions**

* **components/services/models**: kebab-case (e.g., test-component.ts)
* **interfaces/classes**: PascalCase.ts (e.g., Question.ts)

**E. Deployment Checklist**

1. [x] Run tests: npm test
2. [x] Build production: npm run build
3. [x] Check bundle size
4. [x] Test trên nhiều browsers
5. [x] Verify responsive design
6. [x] Check accessibility
7. [x] Update version number
8. [x] Tạo release notes
9. [x] Deploy to server
10. [x] Monitor for errors

**🌟 KẾT LUẬN**

Tài liệu này cung cấp hướng dẫn chi tiết về dự án **TOEIC Writing Test**. Đây là cơ sở để các developers có thể hiểu rõ cấu trúc, luồng hoạt động, và duy trì/phát triển ứng dụng một cách nhất quán.

Để biết thêm chi tiết về chức năng các nút, vui lòng tham khảo file BUTTON\_FLOWS.md trong thư mục docs/.

**© 2025 TOEIC Writing Test Project. All rights reserved.**