POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Technology Report

WO CIAO YOGA - https://yoga-782.pages.dev/

Authors:

TIANQI HU 10948367
SHINUO YAN 11063707
JIANWEI DENG 10973015

Delivery date: **20-06-2025**

Course name: **Hypermedia Applications (Web and Multimedia)**

Academic Year: 2024-2025

Abstract

WO CIAO Yoga is a responsive, modular website for yoga learning and activities, developed as part of the Hypermedia Summer School project. It showcases yoga teachers, class events, daily schedules, and spotlights key courses, aiming to promote yoga as an activity theme. The platform demonstrates the powerful capabilities of frontend frameworks and modern backend solutions in building meaningful digital platforms.

Contents

1 | Introduction

1.1. Group Information

Group name: **WO CIAO**

Group Components

Name Surname	**Person Code**
TIANQI HU	10948367
SHINUO YAN	11063707
JIANWEI DENG	10973015

Table 1.1: Tea Green's components

1.2. Project Information

Link to our website: **https://yoga-782.pages.dev/**

Link to our GitHub repository: **https://github.com/Simon0619/WO_CIAO_YOGA**

1.3. Work Breakdown

We have tried the best we could to keep the work balance among all members as equal and as widespread as possible, in order to let everyone learn the most while doing this project. However, to speed up our work, we had to split the workload at some point: specifically, Every person has contributed to the overall design and development of all aspects of the website. For the pages,we have divided the workload as follows SHINUO YAN: Home Page,About Page, TIANQI HU:Teachers Page,Contact page, JIANWEI DENG:Activities Page,Schedule Page. We advanced the project by defining common development guidelines, allowing each member to touch all aspects of website development, from design to the implementation of backend components. The specific division of tasks is as follows: TIANQI HU:Focused on component development and CSS styling and writing parts of the report. JIANWEI DENG: Focused on page implementation and database queries and writing parts of the report. SHINUO YAN: Focused on page and component implementation and integration and writing parts of the report.

2 | Documentation

2.1. Chosen Theme

We have decided to create a website for a company called "**WO CIAO Yoga**", based in Shanghai, CHINA. We believe in the power of yoga to help people lead happier, healthier and more balanced lives. With a community of dedicated and passionate teachers across Asia and North America, Hydrogen Yoga offers exceptional teaching, facilities and environment that inspires people of all ages and abilities to make yoga part of their lives.

2.2. Technological Choices

- Server-side Development: Nuxt 3 Server Engine (Nitro). We utilized Nitro's capabilities to create server API endpoints directly within our Nuxt project. This integrated approach simplified development, allowing us to manage both frontend and backend logic in a unified codebase.

- Application Hosting: Cloudflare Pages. Chosen for its seamless integration with Git, automatic deployments, and global CDN, which ensures fast delivery of our statically generated site to users worldwide.

- Database Hosting: Cloudflare D1. As a serverless, SQLite-based database, D1 was the perfect fit for our project's needs. Its lightweight nature and ease of use are ideal for structured, small-to-medium scale applications, avoiding the complexity of larger database systems.

- Rendering Mode: Static Site Generation (SSG). We opted for SSG to pre-render all pages at build time. This strategy provides superior performance with near-instant page loads and enhances SEO by serving fully-formed HTML to web crawlers.

- CI/CD and Build: Nuxthub. We used Nuxthub to automate the deployment process and simplify versioned builds, streamlining the path from development to production on Cloudflare.

- Scripting Language: TypeScript. We used TypeScript to enforce type safety throughout the application. This was particularly beneficial when defining data structures for our database entities with Drizzle ORM and component props, leading to more robust, bug-free, and maintainable code.

2.3. Project Structure

2.3.1. Links/Pages Structure

Links/Pages

Page	URL	Description
Home	/	The main landing page, featuring a welcome message and a highlighted section for important classes or events.
Teachers	/teacher	Displays profiles for all yoga teachers, showcasing their résumés and experience.
Person	/persons/[id]	
Activities	/activities	Lists detailed information for all available yoga classes and courses offered by the studio.
course	/activities/[id]	
Schedule	/schedule	Presents the daily and weekly timetable for all yoga classes, allowing users to plan their attendance.
About	/about	A static page providing information about the WO CIAO Yoga project and its mission.
Contact	/contact	Provides contact information for the yoga studio.

Table 2.1: WO CIAO YOGA's links/pages

2.3.2. Available Server Endpoints

Server Endpoints

API call	Parameters	Description
/api/teachers	none	Returns the chatbot’s response to the user’s given utterance, taking into account the previous context.
/api/events	none	Returns a subset of data about all the persons involved with SheRise, in order to build cards to show.
/api/chatbot	message: string	Returns all the data about a person given their ID.

Table 2.2: WO CIAO YOGA’s server endpoints

2.4. Custom Types

The database is designed to manage information about teachers, events (courses and special activities), and the relationship between them.

The core relationship is many-to-many: a teacher can teach multiple events, and an event can be taught by multiple teachers. This is managed through a junction table event_teachers.

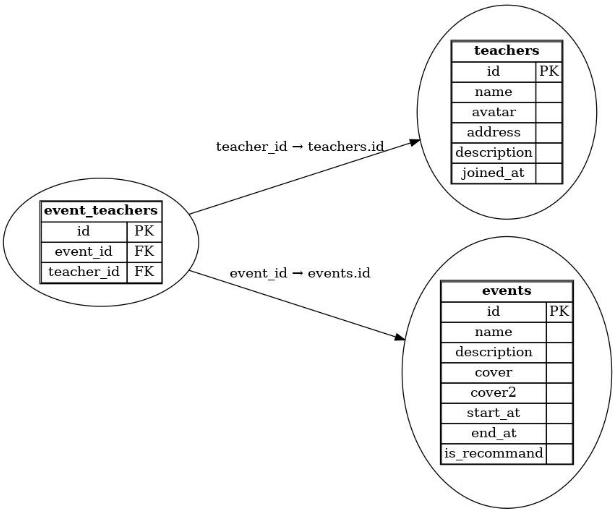

Figure 2.1: Database ER Diagram

Teachers

Stores the basic information for each yoga teacher.

Field Name	Parameters	Description
id	INT	Primary Key, auto-incrementing teacher ID.
name	VARCHAR	Teacher's name.
avatar	VARCHAR	URL for the teacher's profile picture.
address	VARCHAR	Location where the teacher primarily instructs.
description	TEXT	Teacher's personal biography and courses taught.
joined_at	DATETIME	The date the teacher joined the platform.

Table 2.3: teachers table details

Events

Stores information about all courses and recommended activities. A boolean flag distinguishes between regular courses and highlighted events.

Field Name	Parameters	Description
id	INT	Primary Key, auto-incrementing event/course ID.
name	VARCHAR	Name of the event or course.
description	TEXT	Detailed description of the event or course.
cover	VARCHAR	URL for the primary cover image (small)
cover2	VARCHAR	URL for the secondary display image (large).
start_at	DATETIME	Start time of the event/activity.

end_at	DATETIME	End time of the event/activity.
is_recommand	BOOLEAN	Flag to identify a recommended highlight (1=Recommended, 0=Course).

Table 2.4: events table details

Event_teachers

Field Name	Parameters	Description
id	INT	Primary Key, auto-incrementing.
event_id	INT	Foreign Key, references events(id).
teacher_id	INT	Foreign Key, references teachers(id).

Table 2.5: event_teachers table detail

2.5. Custom Components

Our website is constructed from a series of reusable and modular Vue components, designed for clarity and maintainability.

2.5.1. Cards

TeacherCard

This component displays a summary of a yoga teacher, including their photo, name, and primary specialty. It is used on the /teacher page.

TeacherCard

Prop	Type	Required	Short Description
teacher	Object	Yes	An object containing the teacher's data (name, photoUrl, specialty).

Table 2.3: TeacherCard detail

ClassCard

A card component used to display a brief overview of a yoga class or event, including an image, title, and a short description. It is used on the /activities page.

ClassCard

Prop	Type	Required	Short Description
event	Object	Yes	An object with class details (name, imageUrl, summary).
linkTo	String	Yes	The URL path to the detailed page for the class.

Table 2.4: ClassCard detail

2.5.2. Carousels

These are larger components that structure the layout of a page or a significant section of it.

ScheduleView

This component is responsible for fetching and rendering the daily yoga schedule in a clear, tabular format. It is the main component on the /schedule page.

ScheduleView

Prop	Type	Required	Short Description
date	String	Yes	The date for which to display the schedule

Table 2.5: ScheduleView details

PageHeader

A container used at the top of most pages to display a consistent title and an optional introductory text.

PageHeader

Prop	Type	Required	Short Description
title	String	Yes	The main title of the page.
subtitle	String	No	An optional subtitle or introductory sentence.

Table 2.6: PageHeader details

2.5.3. Other Components

This section covers essential components that don't fit into the previous categories. They require no props.

NavBar

This is the main navigation bar for WO CIAO Yoga, present on every page. It includes links to all primary sections of the site and is fully responsive, adapting its layout for mobile and desktop screens.

Footer

The site-wide footer, containing links to social media, contact information, and copyright details.

Chatbot

This component integrates the Tidio chatbot service. It renders a floating button that, when clicked, opens a chat interface, allowing users to interact with a conversational agent.

2.6. Extra Modules

We integrated several external Nuxt modules and libraries to enhance functionality and streamline development.

Extra Modules

Module Name	Usage Description
@nuxt/ui	Provides a comprehensive set of pre-styled, accessible, and themeable UI components that formed the foundation of our site's design system.
drizzle-orm	A type-safe ORM used to interact with our Cloudflare D1 database. It allowed us to write SQL queries in TypeScript, ensuring data consistency.
drizzle-kit	The command-line utility for Drizzle ORM, used for generating and managing database migrations.
csv-parser (dev)	A development-only utility used to parse CSV files for seeding the local database with initial teacher and class data.
tidio	A third-party library for integrating a feature-rich chatbot.

Table 2.7: Extra Modules usage description

2.7. External Libraries

Our project relied almost exclusively on the @nuxt/ui module for styling and components. We did not find the need to import large external CSS libraries like Bootstrap, as Nuxt UI provided sufficient flexibility and a consistent design language out-of-the-box.

3 | Extras

In this chapter, we will discuss how our website is compliant with accessibility and SEO guidelines. To evaluate compliance, we used standard web evaluation tools like Lighthouse and browser accessibility inspectors.

3.1. Accessibility Compliance

The following accessibility good practices have been implemented:

- Alternative Text: Every meaningful non-textual content, such as images of teachers and classes, has a descriptive alt attribute to ensure screen reader users can understand the content.

- High Contrast: The website's color palette, provided by Nuxt UI and customized by us, was chosen to meet WCAG AA contrast ratio standards, ensuring text is clearly readable against its background.

- Clear Structure: We used semantic HTML5 tags (<header>, <main>, <nav>, <footer>, <section>) and a logical heading hierarchy (<h1> to <h4>) to structure content clearly.

- Keyboard Navigation: The entire website is navigable using only a keyboard. Interactive elements like links and buttons have clear focus states.

- Simple Language: The content is written in clear and simple language, avoiding technical jargon to make it accessible to the widest possible audience.

- Standards Compliance: The generated HTML and CSS are standards-compliant and render correctly across all modern web browsers.

3.2. SEO Guidelines Compliance

The following Search Engine Optimization (SEO) good practices have been implemented:

- Unique Title Tags and Meta Descriptions: Each page has a unique, descriptive <title> tag and a meta description that accurately summarizes its content, improving click-through rates from search engine results pages (SERPs).

- Clear Content Hierarchy: The proper use of headings helps search engines understand the structure and importance of the content on each page.

- Clean and Semantic URLs: The URLs are human-readable and contain keywords relevant to the page's content (e.g., /teacher, /activities/vinyasa-flow), which is beneficial for SEO.

- Fast Load Times: By using Static Site Generation (SSG) and deploying on Cloudflare's global CDN, we achieved excellent performance scores, a critical ranking factor for search engines.

- Mobile Friendliness: The website is fully responsive and adapts its layout for optimal viewing on all devices, from mobile phones to desktops.

- Crawlability: A sitemap.xml and robots.txt were generated to help search engine crawlers efficiently discover and index all the important pages of the site.

List of Tables