PROJECT REPORT: HEALTH MANAGEMENT SYSTEM STUDENTS (SHMS)

I. Project Description

1. Application/Website Name

The application is called Student Health Management System (SHMS). It's a web platform that supports the storage, management, and processing of student health data in secondary schools, using a smart, modern, and secure approach.

2. Purpose of the topic

The system's goal is to build an intelligent platform that supports schools in comprehensively managing student health information, including height, weight, BMI, and routine medical history. Health data is automatically updated, accurately processed, and systematically organized, allowing schools to track each student's physical development in detail and continuously.

In particular, the system provides comprehensive data analysis and support for each class and grade level, helping school administrators and homeroom teachers quickly grasp the overall health situation, compare development levels between student groups, and thus promptly detect abnormalities or noteworthy trends. This information will serve as the basis for the school to develop health care plans, organize nutritional counseling, or implement physical training programs suitable for each group of students.

According to data from the World Health Organization (WHO, 2023), the rate of overweight or obese children aged 5–19 globally has reached 20.6%, more than double the rate in 2000. This shows that regular and accurate health management and monitoring of students is extremely necessary in school health care.

The application of technology to health management not only saves human resources but also ensures the accuracy and transparency of data and improves the quality of healthcare for students in a modern educational environment.

3. Reasons for choosing the topic

Currently, in many schools in Vietnam, student health data is still stored manually, recorded on paper, or managed haphazardly in small, fragmented electronic files, lacking synchronization. According to a report by the Vietnamese Ministry of Education and Training (2022), up to 65% of schools have not yet adopted synchronized student health management software.

In the context of digital transformation being an inevitable trend in the field of education, building a professional and modern student health management system will contribute to improving management efficiency, ensuring student health, and meeting the requirements of school administration in the new era.

4. Target users

The application is designed for various stakeholders within the school. The school administration can use it to monitor and manage overall student health data across the school. Homeroom teachers have a tool to quickly check the health status of their students. School health staff can use the system to update routine health check records, analyze data, and plan medical interventions when necessary.

In the future, the application could be expanded to include regular health reports for parents, enabling families to proactively collaborate with the school in caring for and improving their children's health.

II. Description of the idea and main functions:

2.1. Overall Idea

In the context of schools increasingly focusing on monitoring and caring for students' health, building a digital health management system is a necessary and beneficial step. The project "**Student Health Management System**" was created to address that need.

The system is developed on a web platform, using PHP language combined with an SQLite database to optimize for easy deployment on personal computers or the school's internal network without requiring a complex server. The interface is divided into two main user groups:**administrator** and **student**Each user has different access rights, ensuring data security and privacy.

Beyond simply storing personal information and health metrics such as height, weight, and BMI, the system integrates filtering, searching, and data export functions, and allows students to proactively submit correction requests when information is inaccurate. All of this is geared towards achieving a common goal.**Simplify operations – optimize efficiency – ensure information transparency**.

2.2. Main Functions

1. Register and log in

- Allows users to create accounts with the following privileges:admin or student.
- Permissions are set after login: students can only view their own data; administrators have full administrative rights.
- The password is encrypted and secured usingpassword_hash().

2. Student Profile Management (admin)

- Admins can add new students, edit, or delete student information based on student ID.
- You can view a complete list of all students, including their class, date of birth, and identification code.

3. Health data management (admin)

- Enter, update, and delete health information for each student: height, weight, and examination date.
- System **automatically calculate BMI** and **condition classification**according to WHO standards:
    - Under 18.5: Underweight
    - 18.5–24.9: Normal
    - 25–29.9: Overweight
    - From 30 years old and up: Obese
- Automatically record the history of operations (create/update/delete) in the table.history.

4. Student's personal interface

- After logging in, students will be taken to their individual dashboards.
- Personal profiles, medical check-up history, BMI, and health status can be viewed.
- You can submit suggestions to correct height or weight information if you find any errors, using the suggestion form.

5. Manage edit requests (admin)

- Admins can view all suggestions from students.
- You can review, reject, or update the suggested value.
- Proposal status: pending, approved, rejected.

6. Search for students

- The admin dashboard now offers a search feature by student name.
- Update results**real-time**by using JavaScript and the fetch API.

7. Filter by BMI and class

- Allow the admin to filter students by:
    - Tình trạng BMI: Underweight, Normal, Overweight, Obese
    - Class: 6A, 6B, 7A,...

- The filtering results are displayed in the main data table, making it easy for teachers to analyze and track the data.

8. Export CSV data

- The feature allows exporting the entire student health list to a file..csvFor convenient storage, printing, or emailing.
- The data includes: full name, class, height, weight, BMI, BMI status, person who updated the data, and date of examination.

9. Statistics – Charts

- Statistical chart showing average height by class and grade level.
- Use the Chart.js library to visualize data.

2.3. Summary of Key Highlights

- **Clear delegation of authority**The collaboration between administrators and students helps to ensure information security.
- **Simple interface**Easy to use and easy to deploy for schools.
- **Automatic BMI calculation**Accurate classification according to international standards.
  Edit suggestion feature Enhancing transparency and student initiative.
- **Filtering, searching, and CSV export tools**It supports the quick and accurate management and reporting of health information.

III. Database Design

The system's database is designed to ensure:

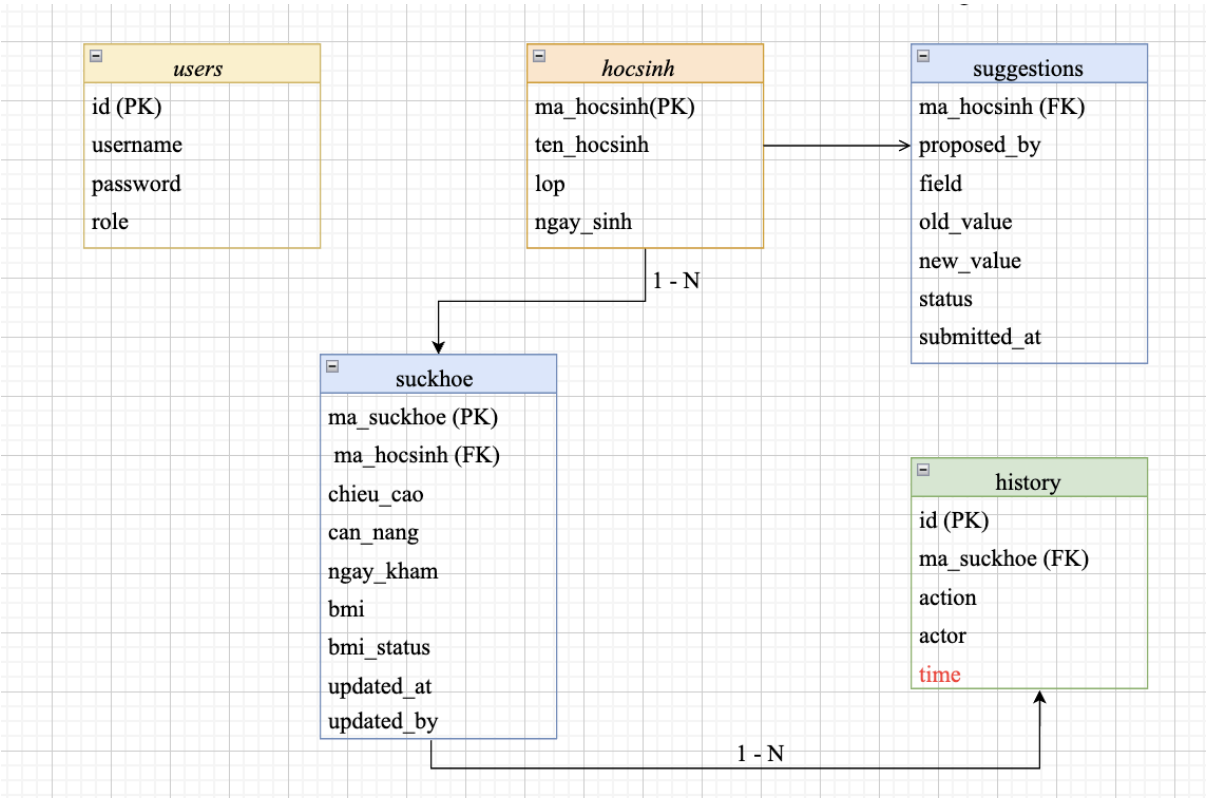
- Manage student information, health, accounts, and edit history comprehensively.
- Supports access control based on account type (admin / student).
- Allow students to submit suggestions for editing their health information.
- It has the potential for scalability and statistical integration.

System usage**SQLite**– a lightweight database management system that is easy to deploy and suitable for internal web applications that do not require high concurrency processing.

1. **Entity Relationship Diagram (ERD)**

Note:

- student.student_IDlink tosuckhoe.ma_hocsinh(One student has multiple health records).
- suckhoe.ma_suckhoelink tohistory.ma_suckhoe(One health record contains multiple historical actions).
- student.student_IDlink tosuggestions.student_ID(One student submitted multiple revision suggestions).



Illustrative ERD Diagram

Notes

- PK (Primary Key): The primary key – a column that uniquely identifies each record in a table.
- FK (Foreign Key): The foreign key – a column that references the primary key of another table to establish relationships between tables.
- 1 - N (One to Many): One record in the parent table relates to many records in the child table.

Relationships between tables

- hocsinh 1 → N suckhoe: One student can have multiple health check records.
- suckhoe 1 → N history: One health record can have multiple edit/change history entries.

- hocsinh 1 → N suggestions: One student can have multiple information update suggestions.
- users: Manages system accounts; it does not have foreign-key relationships with other tables.

2. Quick summary of tables and columns

Board	Describe	Column	Data type	Note
users	Manage login accounts	id	INTEGER, PK, AUTOINCREMENT	Master key
		username	TEXT, UNIQUE, NOT NULL	Login account
		password	TEXT, NOT NULL	Password
		role	TEXT, NOT NULL	Assign permissions ('admin', 'student')
student	Student information	student_code	TEXT, PK	Student ID
		student_name	TEXT, NOT NULL	Student's name
		lop	TEXT, NOT NULL	Class
		date_of_birth	TEXT, NOT NULL	Date of birth
health	Health records	ma_suckhoe	INTEGER, PK, AUTOINCREMENT	Master key
		student_code	TEXT, FK	Foreign keystudent(student_ID)
		height	REAL, NOT NULL	Height
		can_nang	REAL, NOT NULL	Weight
		examination_date	TEXT, NOT NULL	Examination date
		bmi	REAL	BMI
		bmi_status	TEXT	BMI status

		updated_at	TEXT	Date updated
		updated_by	TEXT	Updater
history	Health Editing History	id	INTEGER, PK, AUTOINCREMENT	Master key
		ma_suckhoe	INTEGER, FK	Foreign keysuckhoe(ma_suckhoe)
		action	TEXT	Act
		actor	TEXT	Performed by
		time	TEXT	Execution time
suggestions	Proposal to edit student information	id	INTEGER, PK, AUTOINCREMENT	Master key
		student_code	TEXT, FK	Foreign keystudent(student_ID)
		proposed_by	TEXT	Proposer
		field	TEXT	The school wants to make corrections.
		old_value	TEXT	Old value
		new_value	TEXT	New values
		status	TEXT (default: 'pending')	Browsing status
		submitted_at	TEXT	Date of submission of proposal

3. Relationships between tables

Power supply board	Target board	Type of relationship	Explain

users	student	**1 -1**	A student account corresponds uniquely to one student.
student	health	**1 - many (1-N)**	One student had multiple health check-ups.
health	history	**1 - many (1-N)**	A single health checkup can result in multiple changes to your medical history.
student	suggestions	**1 - many (1-N)**	One student can have multiple revision suggestions.

IV. How to apply and use the database

4.1. Overview

The system uses an SQLite database to store and retrieve data for all user activities. All operations, from logging in, adding students, entering health information, submitting edit requests, to exporting files, are related to this database.**query (SELECT)**, **Insert New**, **Update** or **Delete**data in the database.

The data structure is designed to be flexible and robust enough to support many complex features while still ensuring ease of understanding and simple implementation.

4.2. Functional database query operations

Function	**SQL query type**	**Implementation details**
Register an account	INSERT	Add new users to the table.users, check usernameno duplicates
Log in	SELECT	Queryusersto check login information and permissions
Add students	INSERT	Save the new student's information to the table.student
Update student information	UPDATE	Edit the information including name, class, and date of birth.student_code
Additional health information	INSERT	Write data to the table.health, and simultaneously calculate BMI and classify it.
Update your health information.	UPDATE	Modify health data and log it to the table.history

Delete health records	DELETE	Delete a row in the tablehealth, while also recording the history of the action.
Submit a suggestion for revisions	INSERT	Record the students' suggestions in the table.suggestions
Review / Reject Proposal	UPDATE + UPDATE	Update tablesuggestionsIf approved, the table will be updated immediately.health
Search for students	SELECT with LIKE	Filterstudent_nameContains search keywords, combined with class or BMI filtering.
Filter students by grade level or BMI.	SELECT with WHERE	Add conditionslop = ... or bmi_status = ...to filter by needs
Get a list of health records.	SELECT with JOIN	Combine tablesstudent and health equal JOINto display data
Statistics by class/grade level	SELECT + GROUP BY	Use AVG(height) and GROUP BY lop/khoito calculate the average by class or grade level
Export CSV file	SELECT + PHP fputcsv()	Query the data and then write it as a CSV file for download.

4.3. Typical queries

- **Query to retrieve student and health information:**

sql
```
SELECT hs.ma_hocsinh, hs.ten_hocsinh, hs.lop, sk.chieu_cao, sk.can_nang, sk.bmi, sk.bmi_status, sk.ngay_kham
FROM student
JOIN suckhoe sk ON hs.ma_hocsinh = sk.ma_hocsinh
WHERE hs.ten_hocsinh LIKE '%linh%';
```

- **Save the history of actions (when the admin makes edits):**

sql
```
INSERT INTO history (ma_suckhoe, action, actor, time)
VALUES (?, 'update', 'admin1', datetime('now'));
```

- **Calculate the average height per class:**

sql
```
SELECT hs.lop, ROUND(AVG(sk.chieu_cao), 2) AS avg_height
```

```
FROM student
JOIN suckhoe sk ON hs.ma_hocsinh = sk.ma_hocsinh
GROUP BY hs.lop;
```

- **Submit your suggested edits:**

sql

```
INSERT INTO suggestions (ma_hocsinh, proposed_by, field, old_value, new_value)
VALUES (?, ?, ?, ?, ?);
```

4.4. How to use CRUD (Create – Read – Update – Delete)

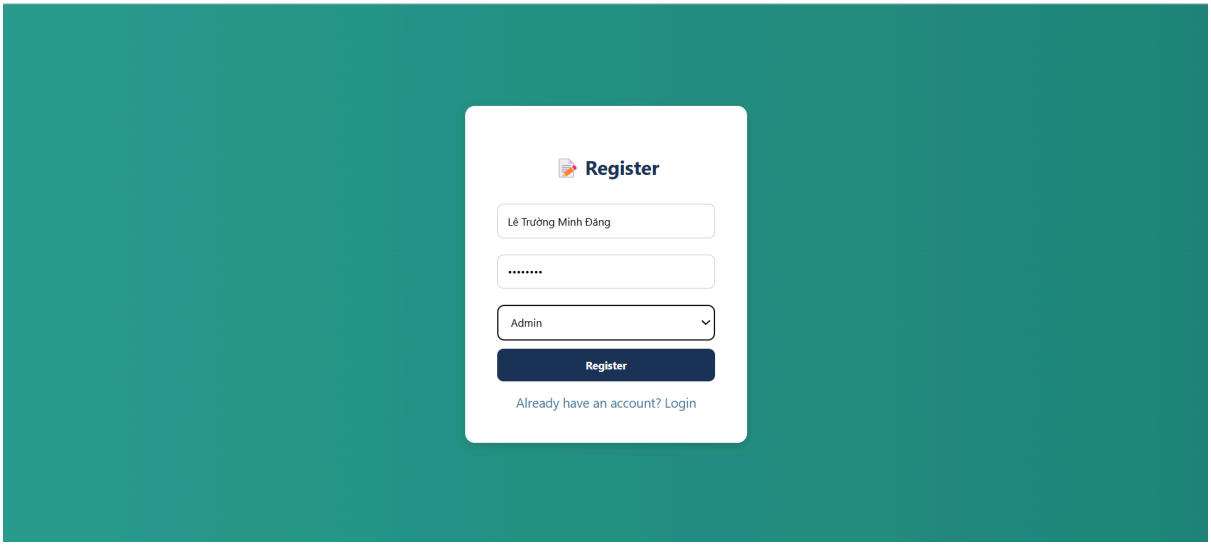
Operation	**Applicable table**	**Real-world context**
CREATE	users, hocsinh, suckhoe, suggestions	When registering, adding students, adding profiles, submitting proposals
READ	All tables	When displaying the dashboard, student list, filter, and search options.
UPDATE	suckhoe, hocsinh, suggestions	When editing profiles, review proposals.
DELETE	health	When deleting student health records

4.5. Scalability and Optimization

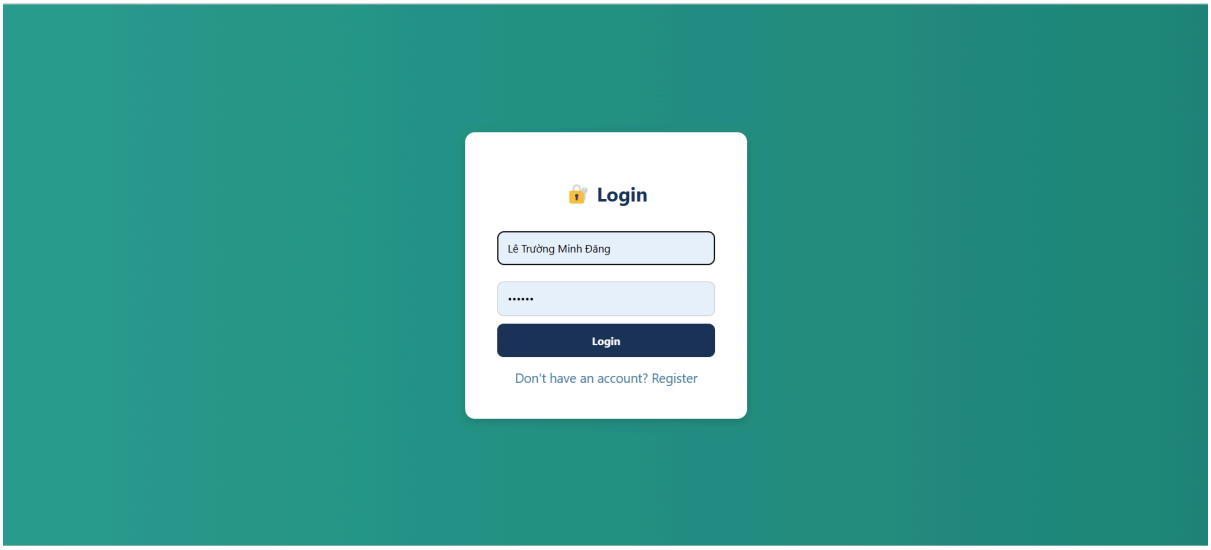
- **BMI**The calculations are performed and stored directly in the database, reducing the processing load on the client side.
- Use **prepared statements (PDO)**Helps prevent SQL Injection.
- It's possible to upgrade the system to MySQL or PostgreSQL without significantly altering the logic.
- The database design follows normalization, but is still easily expandable with additional tables, such asnotifications, classes, teachers,...

V. Results of Implementation

The registration page interface

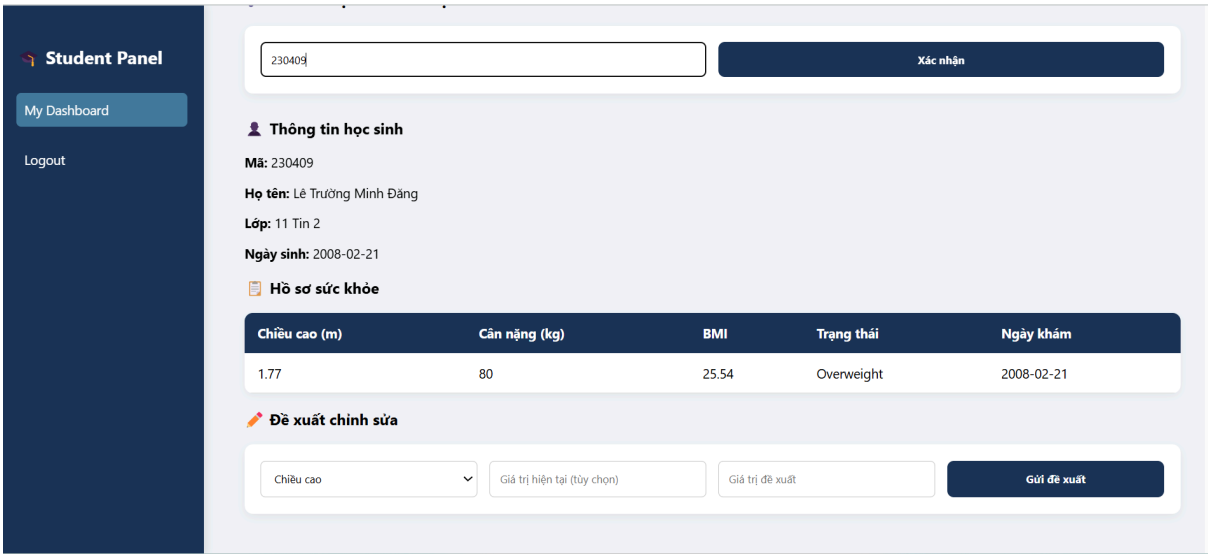


Login page interface

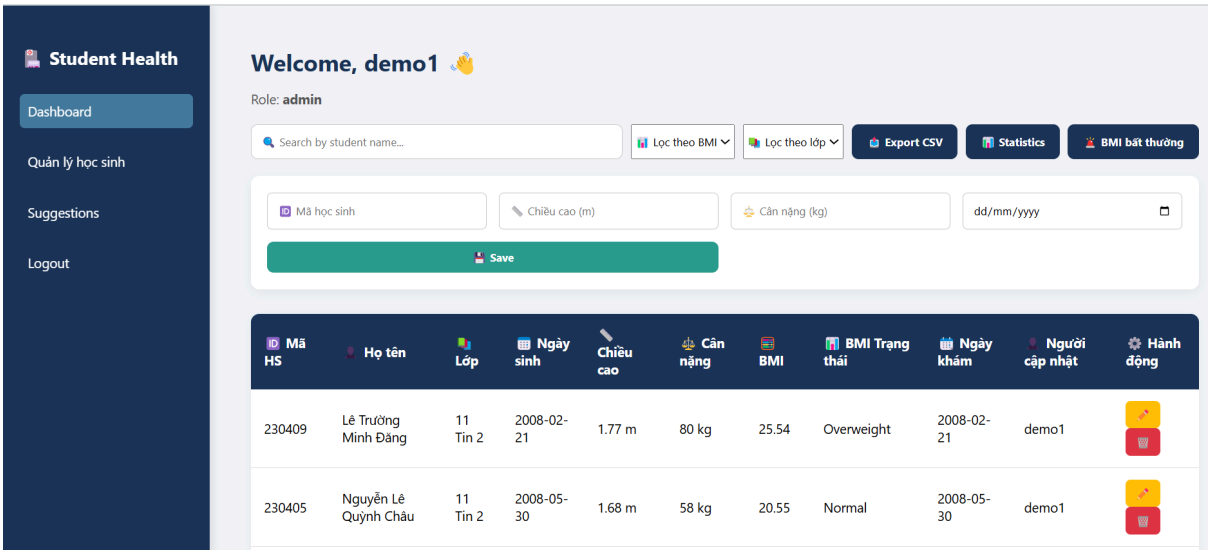


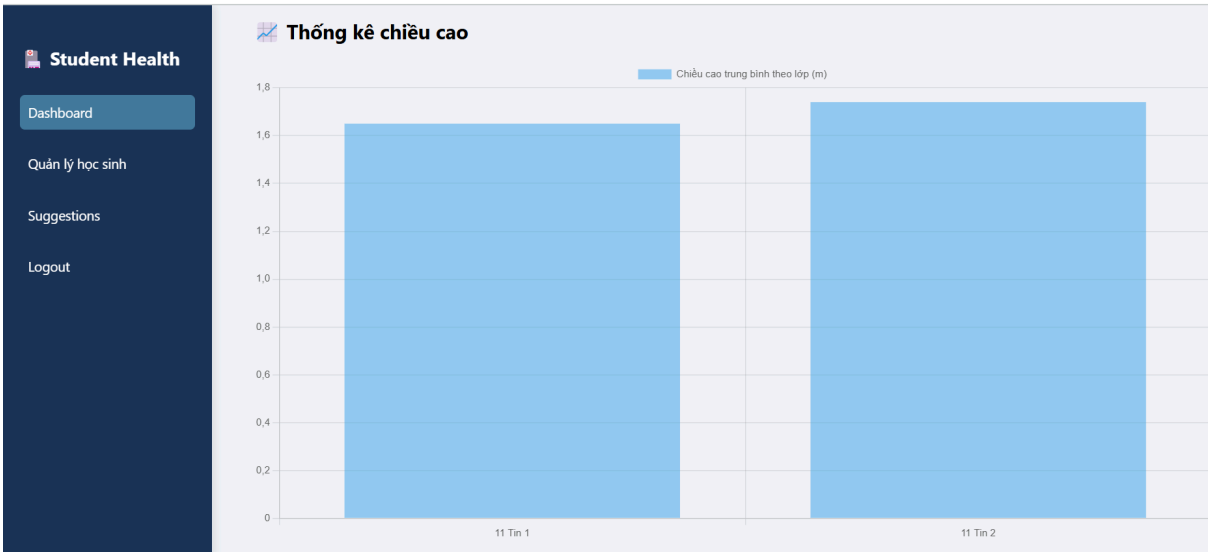
Student role interface

When receiving health updates from the admin.

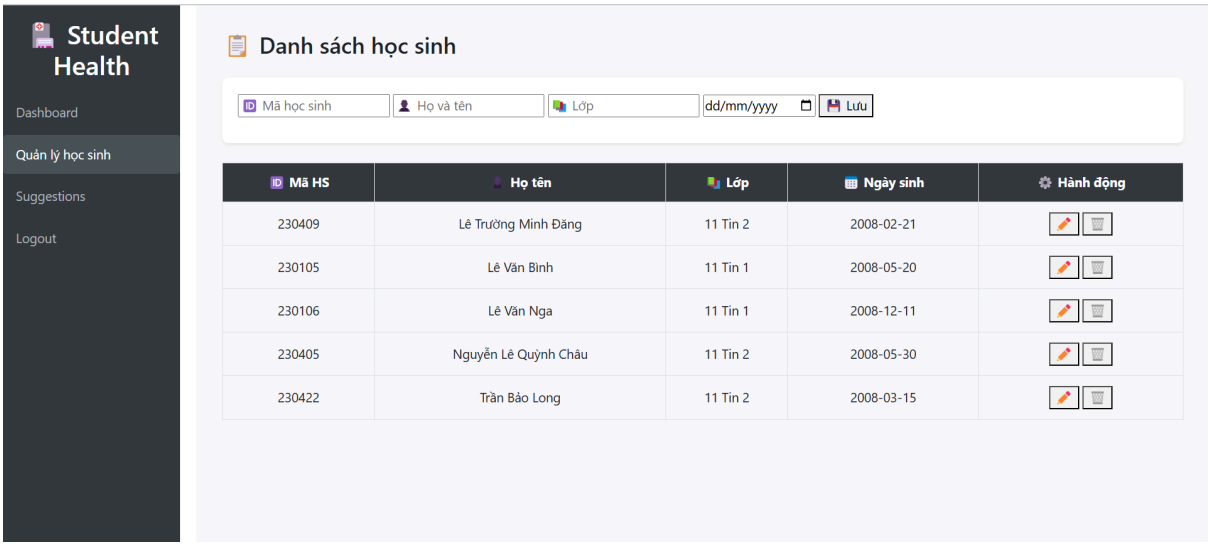


Admin dashboard homepage

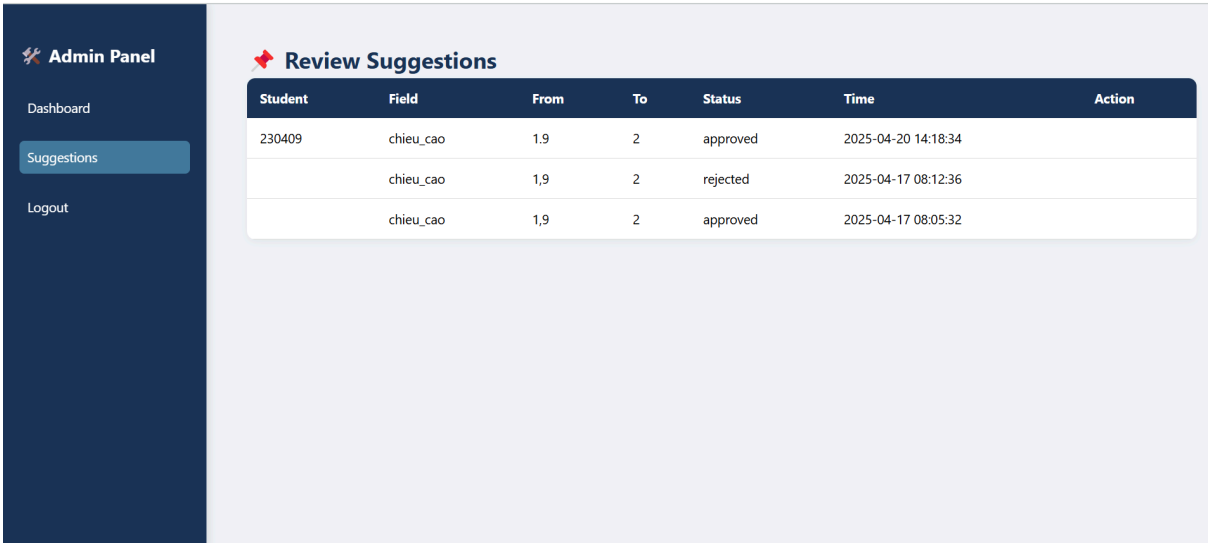




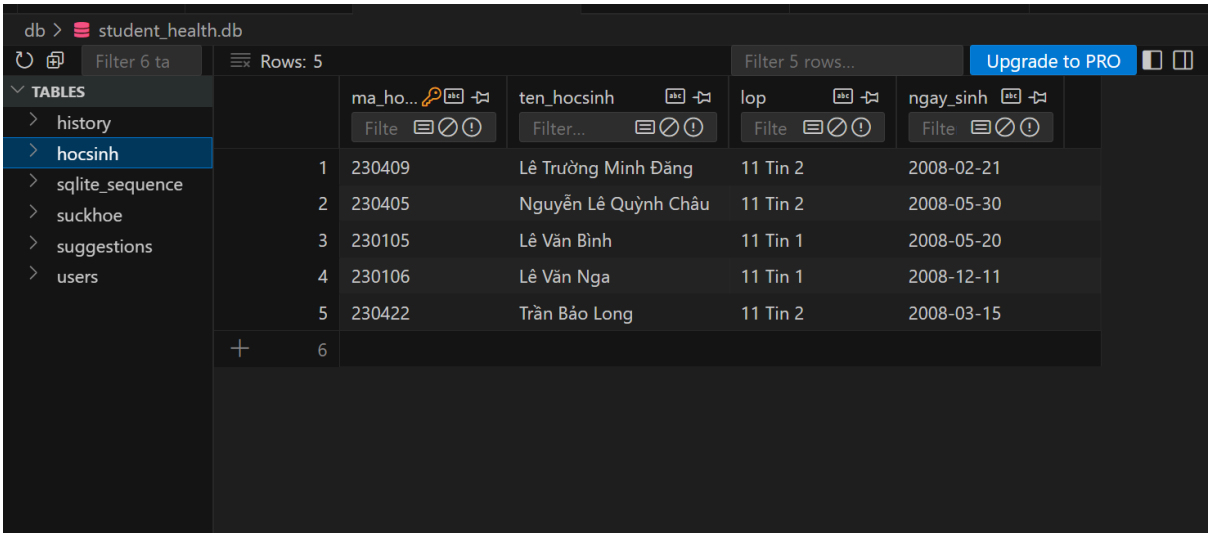
Student list management page



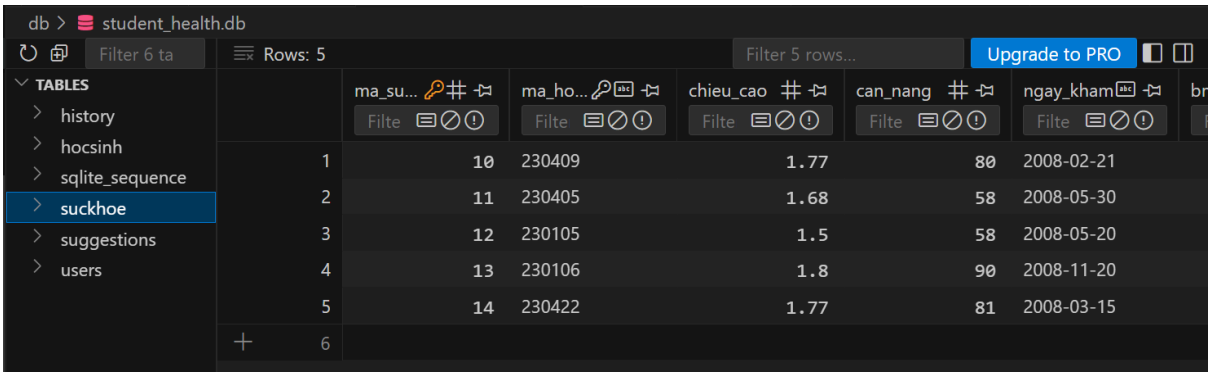
The page allows you to view the student's proposed edits and decide whether to approve or reject the changes.



Student Table



Health Table



db > student_health.db

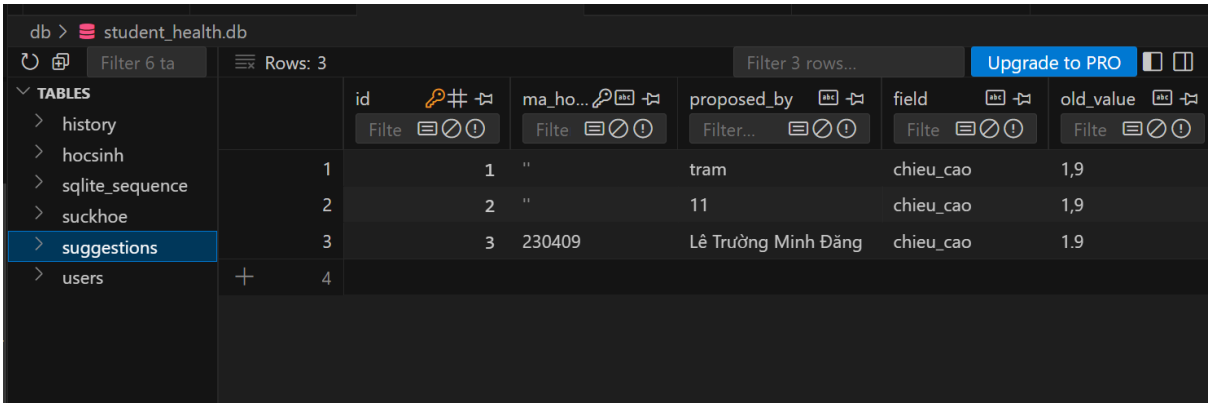
Filter 6 ta | Rows: 5 | Filter 5 rows... | Upgrade to PRO

TABLES: history, hocsinh, sqlite_sequence, suckhoe, suggestions, users

	ma_su...	ma_ho...	chieu_cao	can_nang	ngay_kham	br
1	10	230409	1.77	80	2008-02-21	
2	11	230405	1.68	58	2008-05-30	
3	12	230105	1.5	58	2008-05-20	
4	13	230106	1.8	90	2008-11-20	
5	14	230422	1.77	81	2008-03-15	

6

Proposed table



db > student_health.db

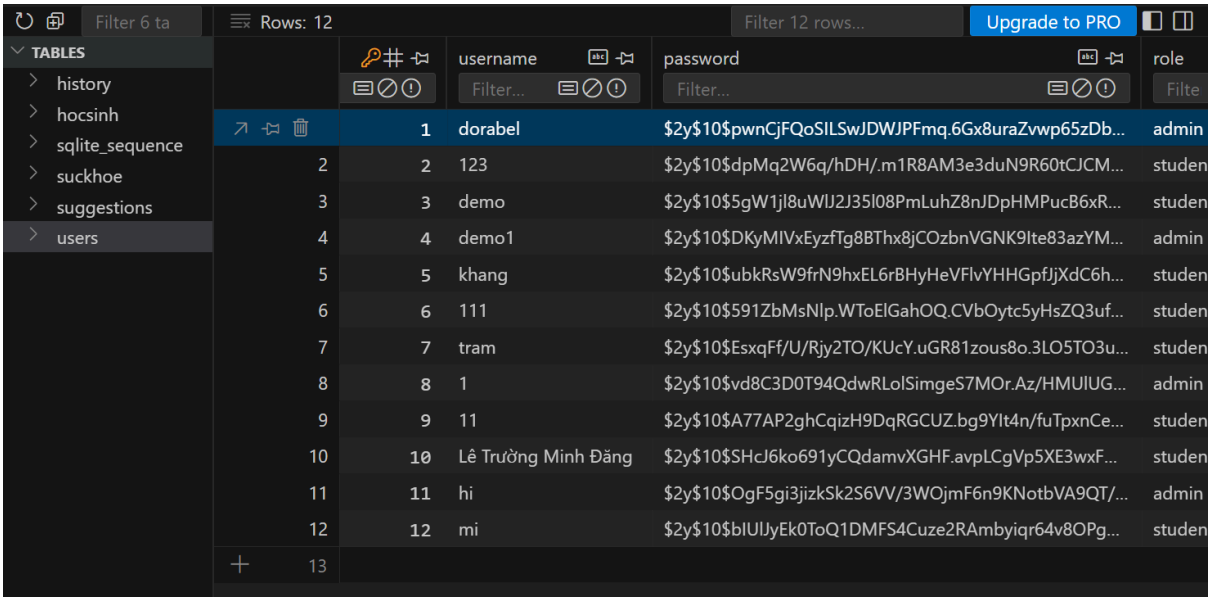
Filter 6 ta | Rows: 3 | Filter 3 rows... | Upgrade to PRO

TABLES: history, hocsinh, sqlite_sequence, suckhoe, suggestions, users

	id	ma_ho...	proposed_by	field	old_value
1	1	''	tram	chieu_cao	1,9
2	2	''	11	chieu_cao	1,9
3	3	230409	Lê Trường Minh Đăng	chieu_cao	1.9

4

Table users



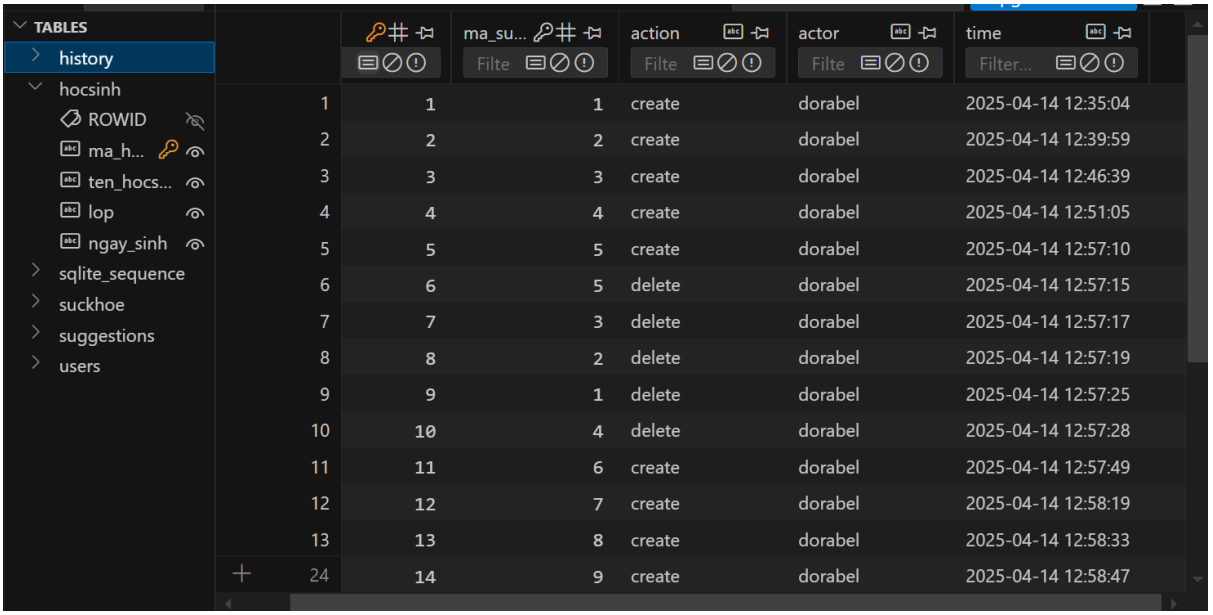
Filter 6 ta | Rows: 12 | Filter 12 rows... | Upgrade to PRO

TABLES: history, hocsinh, sqlite_sequence, suckhoe, suggestions, users

		username	password	role
1	1	dorabel	$2y$10$pwnCjFQoSILSwJDWJPFmq.6Gx8uraZvwp65zDb...	admin
2	2	123	$2y$10$dpMq2W6q/hDH/.m1R8AM3e3duN9R60tCJCM...	studen
3	3	demo	$2y$10$5gW1jl8uWlJ2J35l08PmLuhZ8nJDpHMPucB6xR...	studen
4	4	demo1	$2y$10$DKyMIVxEyzfTg8BThx8jCOzbnVGNK9lte83azYM...	admin
5	5	khang	$2y$10$ubkRsW9frN9hxEL6rBHyHeVFlvYHHGpfJjXdC6h...	studen
6	6	111	$2y$10$591ZbMsNlp.WToElGahOQ.CVbOytc5yHsZQ3uf...	studen
7	7	tram	$2y$10$EsxqFf/U/Rjy2TO/KUcY.uGR81zous8o.3LO5TO3u...	studen
8	8	1	$2y$10$vd8C3D0T94QdwRLolSimgeS7MOr.Az/HMUlUG...	admin
9	9	11	$2y$10$A77AP2ghCqizH9DqRGCUZ.bg9Ylt4n/fuTpxnCe...	studen
10	10	Lê Trường Minh Đăng	$2y$10$SHcJ6ko691yCQdamvXGHF.avpLCgVp5XE3wxF...	studen
11	11	hi	$2y$10$OgF5gi3jizkSk2S6VV/3WOjmF6n9KNotbVA9QT/...	admin
12	12	mi	$2y$10$bIUlJyEk0ToQ1DMFS4Cuze2RAmbyiqr64v8OPg...	studen

13

Table history



		ma_su...	action	actor	time
1	1	1	create	dorabel	2025-04-14 12:35:04
2	2	2	create	dorabel	2025-04-14 12:39:59
3	3	3	create	dorabel	2025-04-14 12:46:39
4	4	4	create	dorabel	2025-04-14 12:51:05
5	5	5	create	dorabel	2025-04-14 12:57:10
6	6	5	delete	dorabel	2025-04-14 12:57:15
7	7	3	delete	dorabel	2025-04-14 12:57:17
8	8	2	delete	dorabel	2025-04-14 12:57:19
9	9	1	delete	dorabel	2025-04-14 12:57:25
10	10	4	delete	dorabel	2025-04-14 12:57:28
11	11	6	create	dorabel	2025-04-14 12:57:49
12	12	7	create	dorabel	2025-04-14 12:58:19
13	13	8	create	dorabel	2025-04-14 12:58:33
24	14	9	create	dorabel	2025-04-14 12:58:47

Link demo video

Login - Student Health - Google Chrome 2025-04-22 22-25-12.mp4

VI. Evaluation & Development Directions

6.1. Difficulties encountered

- **User permission management**: Building two separate interfaces foradmin and studentAlong with restricting access, this requires careful handling of session logic and access checks at all API access points and interfaces.

- **Accurate and automatic BMI calculation**Accuracy must be ensured in height-related divisions, avoiding division by zero errors or incorrect input formats.

- **Interacting with the database**Although SQLite is simple, handling foreign key constraints and updating linked data (JOIN, filtering by class/BMI) still requires a well-structured and efficient query system.

- **Processing statistical charts**The output data needs to be standardized to create charts using the Chart.js library in a way that is intuitive and easy to understand.

6.2. Strengths of the product

- ✅ **Clear, secure, and reliable access control.**Each student can only access their own data; the administrator has full control over the system.

- ✅ **Automatic BMI calculator system**Classified according to WHO international standards, this system is well-suited for monitoring students' health.

- ✅ **Simple, intuitive, and user-friendly interface.**It is easy to use for teachers, students, and school health staff alike.

- ✅ **Smart filtering feature**This allows for quick data retrieval based on class or health status.

- ✅ **Flexible editing suggestions**From the students' perspective – creating a two-way, transparent, and proactive management environment.

- ✅ **Good scalability**Additional features such as notifications, OTP authentication, PDF report generation, or upgrading to a larger database like MySQL/PostgreSQL can be integrated when needed.

6.3. Future Development Directions

1. **Integrate email or internal notifications.**To send health alerts to homeroom teachers, students, or parents.

2. **Export the report as a PDF or Excel file.**based on class, block, or BMI status periodically.

3. **Add real-time statistics functionality.**For example, tracking the average growth rate of students over each examination period.

4. **Support proposal verification with multi-step authentication.**(Multi-step verification) to ensure accuracy during updates.

5. **Deploy a fully mobile or responsive version.**So that students can look up their health information anytime, anywhere.

VII. Instructions for running the application (so that teachers can run the application):

7.1. Software Requirements

- ✅ PHP ≥ 7.4 (latest version recommended)

- ✅ SQLite3 (already built into most PHP versions)

- ✅ Web Server: You can use:

- XAMPP / MAMP / WAMP (on Windows/macOS)
- Alternatively, use the built-in PHP server (run the command).php -S localhost:8000)

No MySQL, Docker, or complex environment configuration is required – the system is suitable for offline environments.

7.2. Installation and Launch Steps

Step 1: Download the source code to your computer.

- Copy the source code to any folder (if using the built-in server) or directory.htdocs(if using XAMPP).

Step 2: Create a database

- Accesshttp://localhost:8000/init_db.php (or http://localhost/project_folder/init_db.php) to create the filestudent_health.db.

Step 3: Start the system

- Accesshttp://localhost:8000/login.htmlto log in.
- Register a new account with the following role.admin or studentAt your discretion.

Step 4: Use

- Admin: Add students, enter health information, view edit history, approve proposals.
- Students: review personal profiles, suggest edits to health data.

References (APA 7th edition)

- World Health Organization. (2023). *Obesity and overweight*. World Health Organization. Retrieved April 22, 2025, from https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweigh
- Ministry of Education and Training of Vietnam. (2022). *Annual education sector review report for the academic year 2021-2022*. Retrieved April 22, 2025, from https://moet.gov.vn/tintuc/Pages/thong-tin-giao-duc.aspx
- Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley.
- Beighley, L., & Morrison, M. (2009). *Head First PHP & MySQL*. O'Reilly Media.
- Owens, M. (2010). *The Definitive Guide to SQLite*. Apress.

- World Health Organization. (2004). *Appropriate body-mass index for Asian populations and its implications for policy and intervention strategies*. The Lancet, 363(9403), 157-163.
- Nielsen, J. (1995).*10 Heuristics for User Interface Design*. Nielsen Norman Group.
- Federal Trade Commission. (2016). *Protecting Personal Information: A Guide for Business*.
- ISO/IEC 27001:2013. (2013). *Information technology — Security techniques — Information security management systems — Requirements*. International Organization for Standardization.
- Robbins, J. N., & Beaird, J. (2017). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics* (5th ed.). O'Reilly Media.
- Rouse, M. (2020). *CRUD (Create, Read, Update, Delete)*. TechTarget.
- Mozilla Developer Network. (n.d.). *Using fetch*.