**Daycare Management System**

**Team Members:**

Shao Hang Li

Ritu Arora

**Qi He\***

**Project Description:**

The Daycare Management System is a web app that streamlines daycare administration and improves communication among parents, educators, and managers. It lets users capture photos during registration for quick child profile uploads and uses dynamic charts to monitor attendance, billing, and performance.

The system also includes a calendar to schedule classes, events, and meetings. User roles (parent, educator, manager) determine which features are accessible. It supports multiple languages, is mobile-responsive with Bootstrap, and incorporates key security measures like email activation, password resets, and reCaptcha.

* **Technologies Used:**
  + PHP Slim framework (or Laravel not sure yet)
  + MeekroDB for database interactions
  + Twig for templating
  + Monolog for logging
* **Special Features:**
  + Image capture during registration
  + Interactive HTML5 charts for analysis
  + Calendar scheduling for classes and events
  + Multi-role access (parent, educator, manager)
* **Additional Libraries:**
  + AJAX for dynamic content updates
  + Facebook login (PHP library, not JS)
  + Google Maps integration
  + ReCaptcha at registration/login
  + Email activation and password reset functionalities
  + Bootstrap for responsive design and jQuery for enhanced interactivity
  + FullCalendar for scheduling and HTML5 canvas for charts
* **Challenging Items Not Covered in Class:**
  + Implementing secure payment processing (Stripe/PayPal in test mode) - Stripe API preferred
  + Integrating camera support for on-the-spot image capture - more of FE

### **Web Pages**

* **/index or /** – Home page that introduces the system and provides navigation.
* **/about** – About page (or section) describing the daycare.
* **/services** – Page listing available services with details.
* **/login** – Page for user login.
* **/register** – Page for new user registration.
* **/profile** – Page for editing a user profile and viewing service subscriptions.
* **/order** – Page with an integrated calendar for placing orders.

Optional: The Analytics table is optional. Analytics can be computed on the fly using queries, and the Messaging table can be omitted if we choose to handle communication through another system (such as email or a third-party messaging service).

**Entity Relationship Design of the database**

Below is a list of your tables along with a brief description for each:

1. Users Table

Purpose: Stores all user information including name, email, password, role (parent, educator, manager), account activation status, and a soft delete flag.

Key Fields: id, name, email, password, role, activation\_status, isDeleted, created\_at.

1. Children Table

Purpose: Holds records for each child enrolled in the daycare and links each child to their parent (a user). Also includes a soft delete flag.

Key Fields: id, parent\_id (FK to users), name, date\_of\_birth, profile\_photo\_path, isDeleted.

1. Registrations Table (or Attendance Table)

Purpose: Logs daily attendance details for children, indicating if a child was present or absent on a specific day. Also includes a soft delete flag.

Key Fields: id, child\_id (FK to children), registration\_date, status (present/absent), isDeleted.

1. Events Table

Purpose: Stores scheduling information for classes, meetings, or other events in the daycare. Also includes a soft delete flag.

Key Fields: id, title, start\_date, end\_date, description, isDeleted.

1. Payments Table

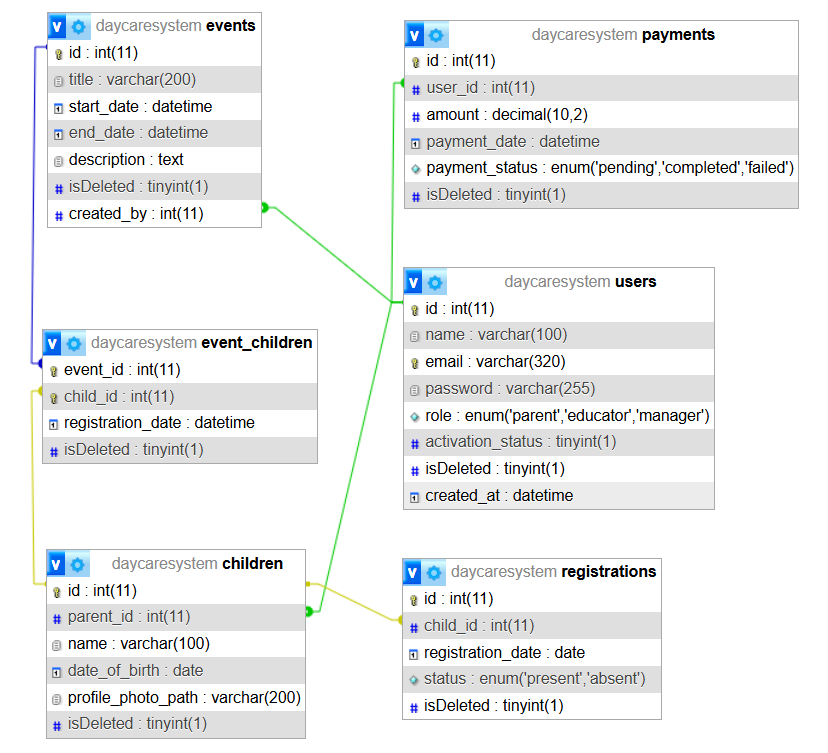
Purpose: Records billing and payment transactions made by users (e.g., for tuition or services). Also includes a soft delete flag.

Key Fields: id, user\_id (FK to users), amount, payment\_date, payment\_status, isDeleted.

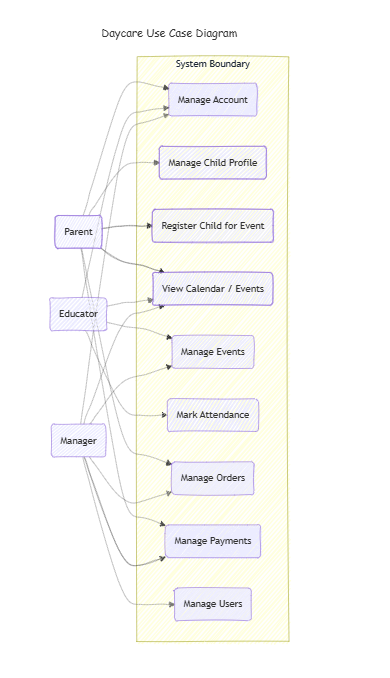
1. Event\_Children Table

Purpose: A junction table that creates a many-to-many relationship between children and events, indicating which children are registered for which events. This table replaces the previous Event\_Attendees table and includes a soft delete flag.

Key Fields: event\_id (FK to events), child\_id (FK to children), registration\_date, isDeleted.



**Use case diagram**

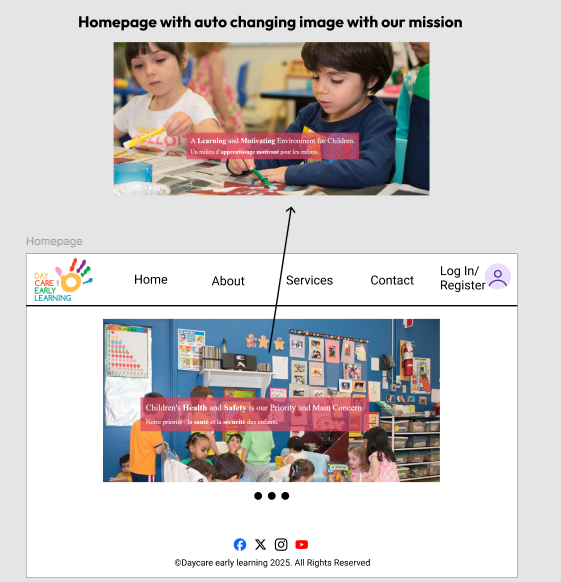
****

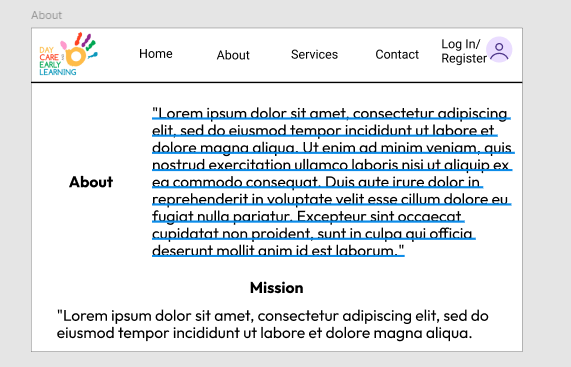
**Mockups**

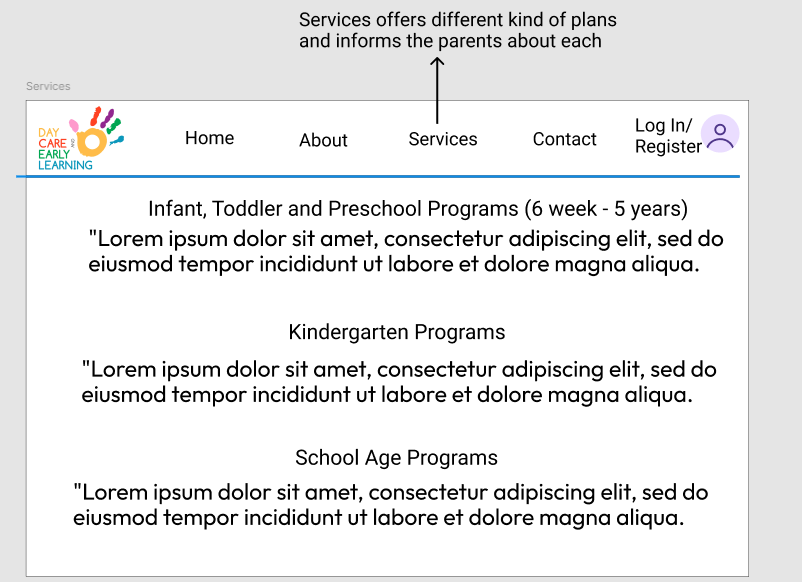
Our design draws inspiration from the following leading daycare management system:  
[C.P.E Concordia | C.P.E Concordia's Website](https://cpeconcordia.ca/)

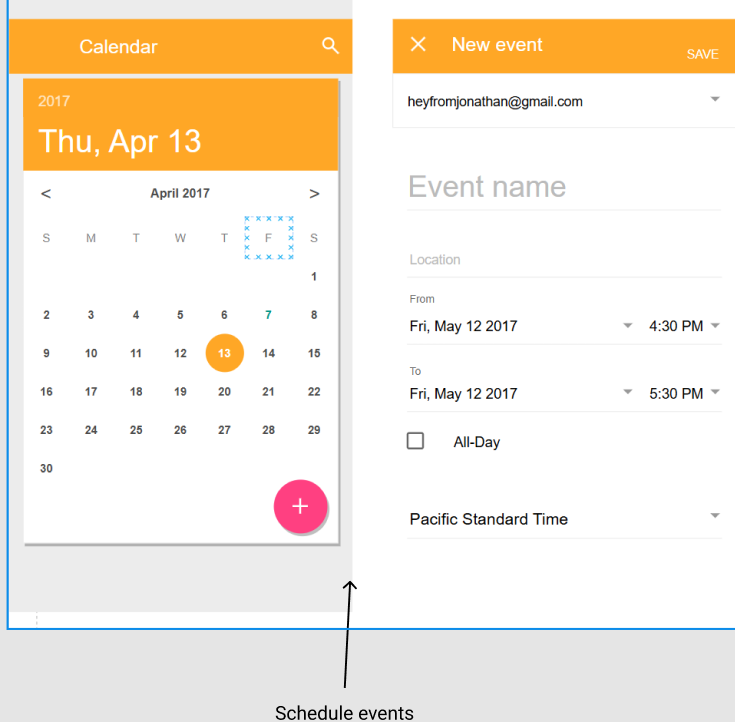
<https://www.laplace0-5.com/>

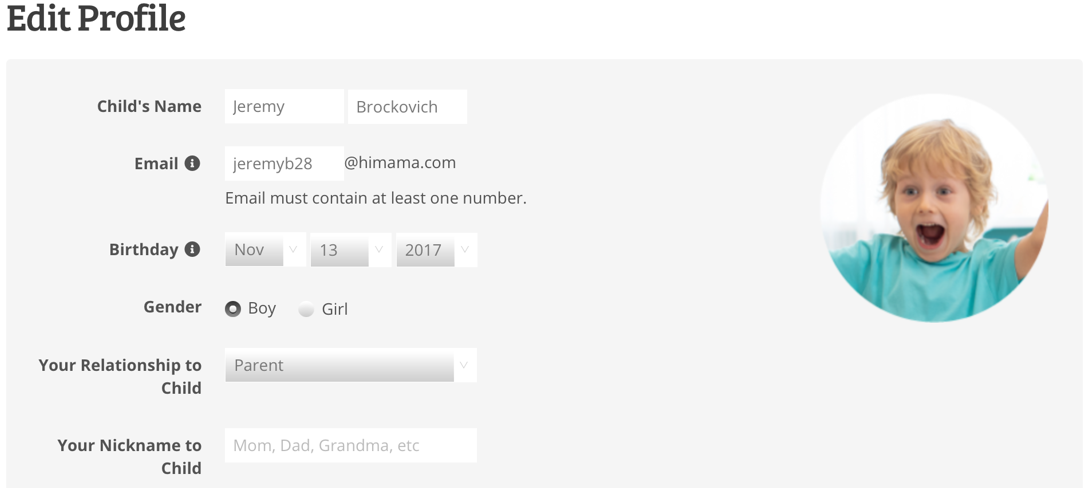
**https://www.figma.com/design/kr1YUfgjxMoKBu2L0LEAo6/Flux---Figma-Build-Tutorial-(Starter)-(Community)?node-id=203-1992&t=UMrbdzkCXn5DWUpP-1**

****

****

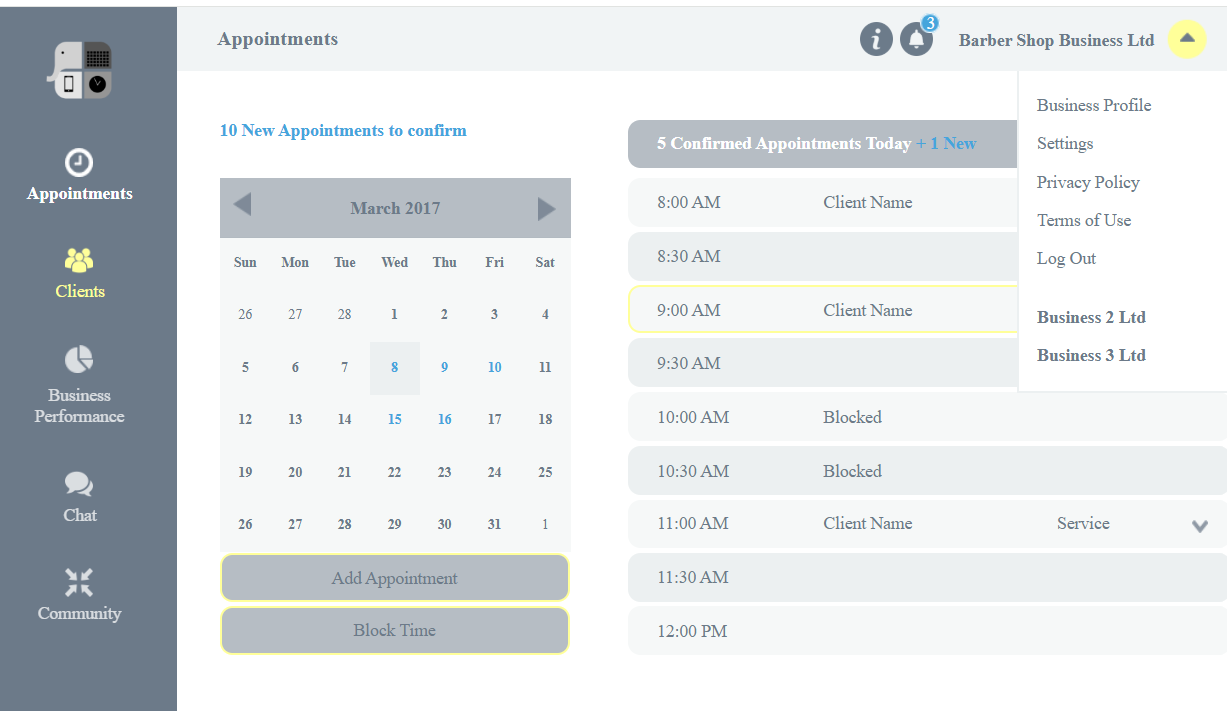
****

****

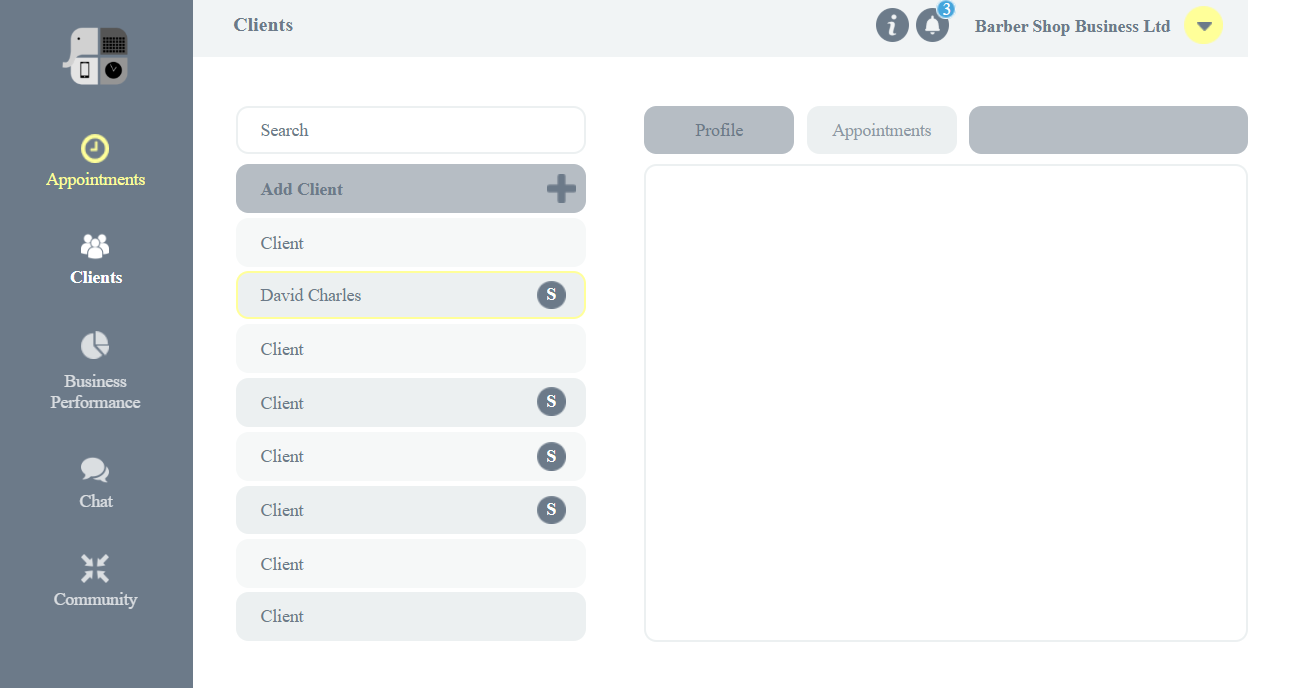
****

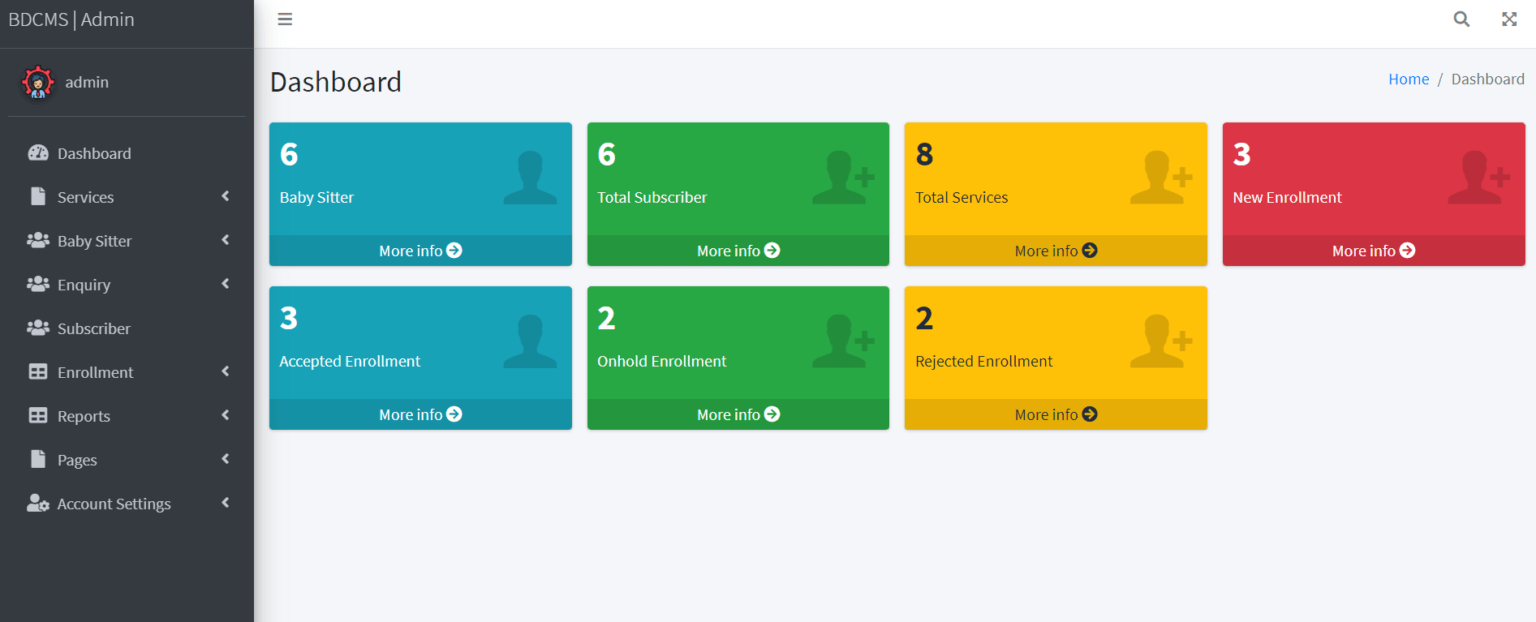
For pictures, validate extension and size and also scale down a picture on upload, UUID for filenames . ext

**Admin Dashboard to manage clients and their appointments, view analytics**

****

list of children registered loaded via AJAX , pagination via AJAX/page no in url

****

****

Using Roles - for user management service - authorization to different functionalities  
Users (parents and educators) managed by Admin