

IPD23 Team8 PHP Project Proposal

Team member: Team8 (@ Ying LUO @ Khalil Hanna)


Ski Equipment Rental Website

Project Description:


The idea to provide online service to rent skiing equipment by a period of season or month.

- Create a website for ski equipment rental
- User management(admin/user registration)
- Equipment management(CRUD operations)
- Rental Rate by equipment & time period
- Cart
- Order and Transaction

Typical Scenario:

 Login as Admin

User Management	Equipment Management	Rental Rate	Order
Add new user	Add new equipment	Add rental rate for new equipment	Search order information by order Id /equipment Id/user Id (View all order list)
Update user information	Update equipment information	Update rental rate for existing equipment	
Search user (View all user list)	Search equipment (View all equipment list)	Search Rental Rate by equipment Id (View all rental rate list)	
Delete user (foreign key restrict constraint of orders table before deletion)	Delete equipment (foreign key restrict constraint of rentalRate table before deletion)		

 Login as Register User

User Management	Equipment Management	Rental Rate	Cart	Order Management
Register new user	Search equipment by 'like' equipment name (View all equipment list)	View rental rate from equipment list page (query by inner join of equipment table and rental rate table)	List of temporary selected equipment and quantity (store in client side/cache? with login user session id)	My own order list (query orders table by userId provided from url)
Update my own user information				

List of URLs:

1. Login
 - a. admin ==>4
 - b. user ==>5
 - c. entry of register new==>2
2. Register New User
 - a. or login ==>1

3. Index(Main Page)
 - a. entry of login==>1
 - b. entry of register new user==>2
 - c. entry of all equipment list==>5
 - d. entry of cart==>7
4. Admin - User List(?necessary or not?)
5. Admin / User - Equipment list
 - a. select equipment and add to cart ==>7
 - b. search/query by '%like%' equipment name
 - c. choose one equipment and review detail ==>6
 - d. change rental rate by equipment id == >8
6. Admin / User - Equipment Detail
 - a. Manage Rental Rate(Admin)
7. User - Order Cart
 - a. add/increase items in cart
 - b. choose items in cart and place order ==>8
8. Order Detail
 - a. confirm and choose payment method and pay(by default mock payment succeed)==>9
 - b. cancel==>7
9. Admin / User - Order history List (filter by user id to display for user page)

Possible enhancement:

- change password or find password by email
- transaction report export pdf or print
- equipment rental feedback with comments
- price changing push notice by email
- Manage the stock
- transaction statistic analysis
- search engine recommendation?

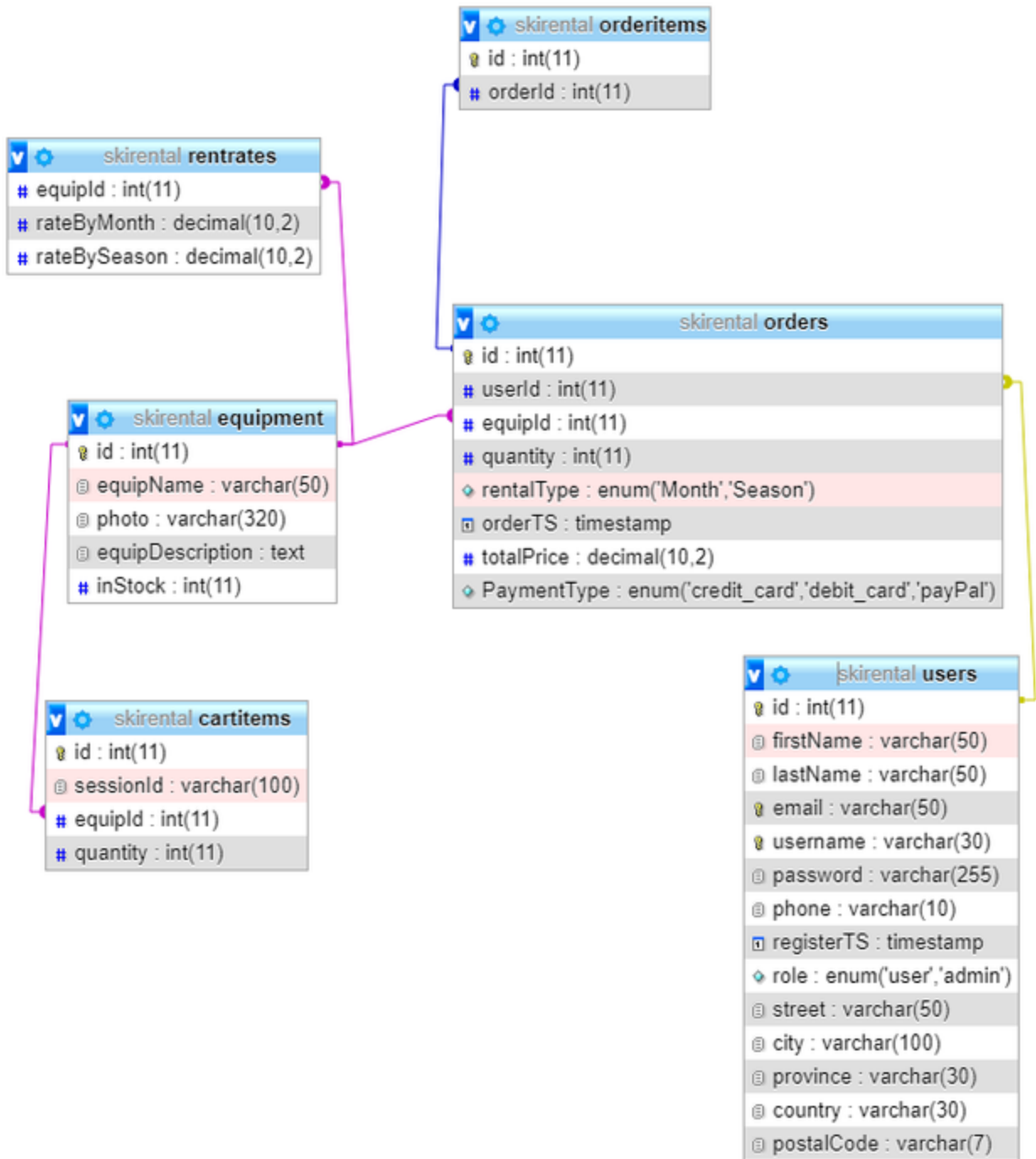
Time Management:

- Total 2 weeks
 - Week1: Basic system design, main pages and CRUD operations implementation
 - Week2: Unit Test, defect fix, incremental features of possible enhancement
-

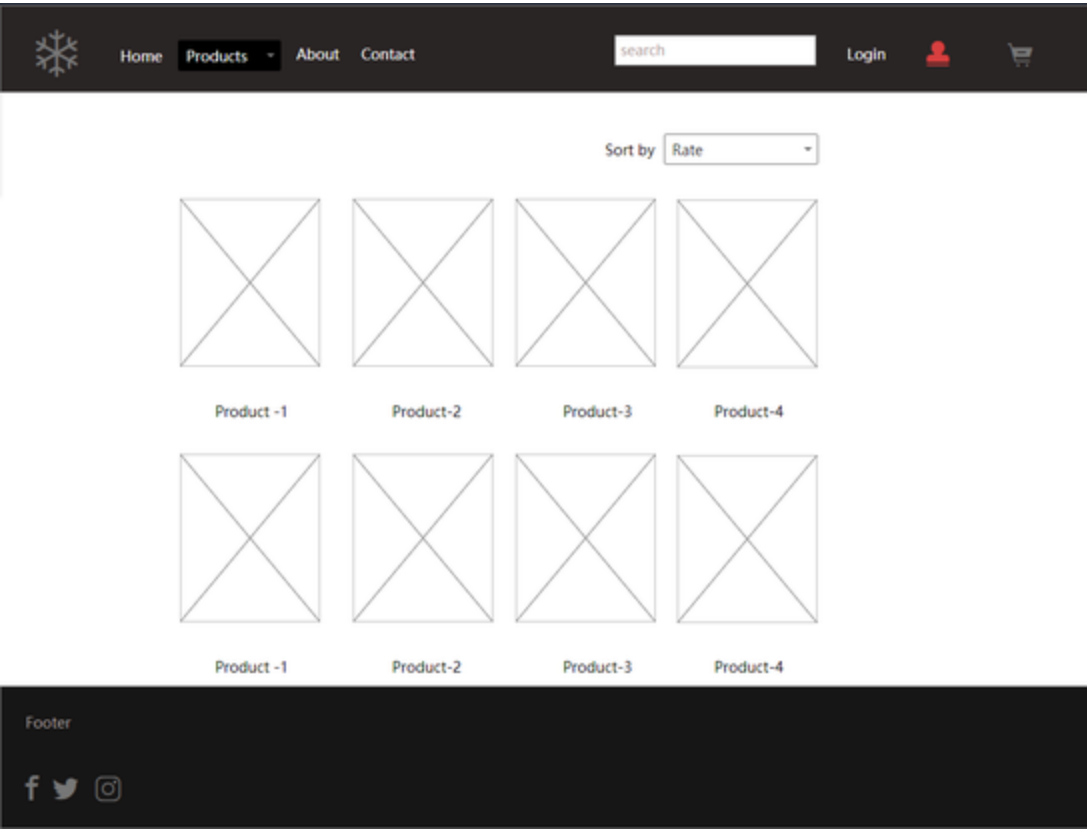
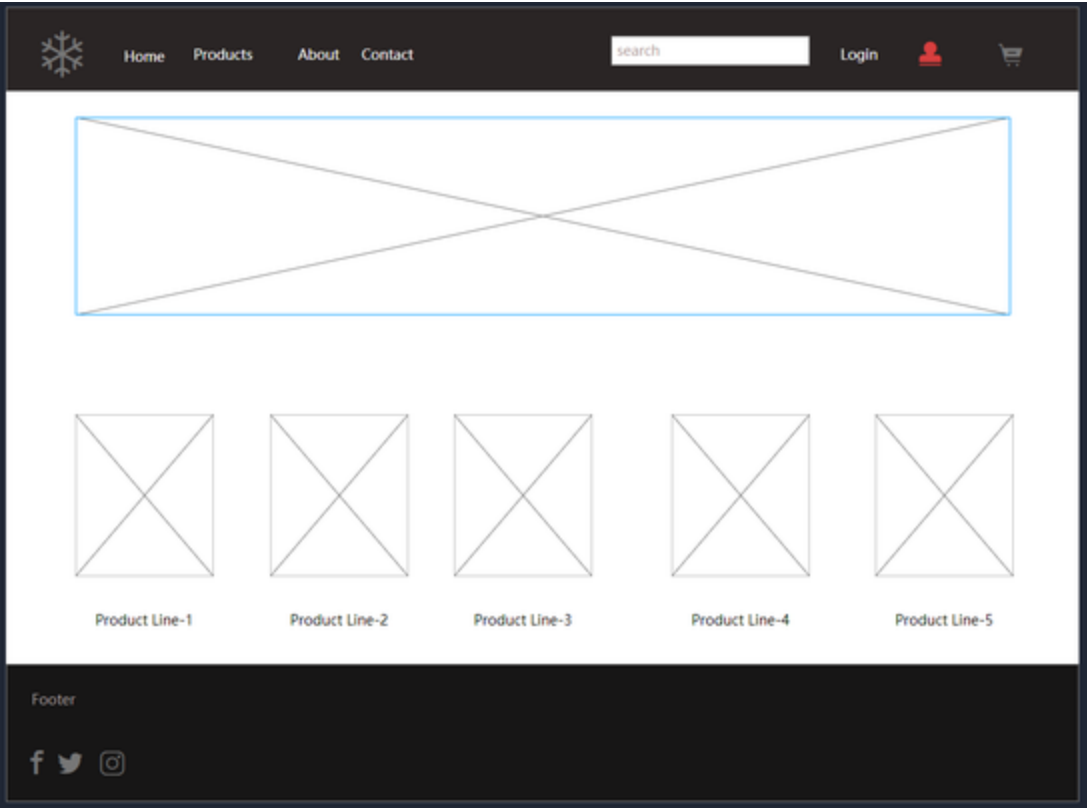
ERD Diagram:

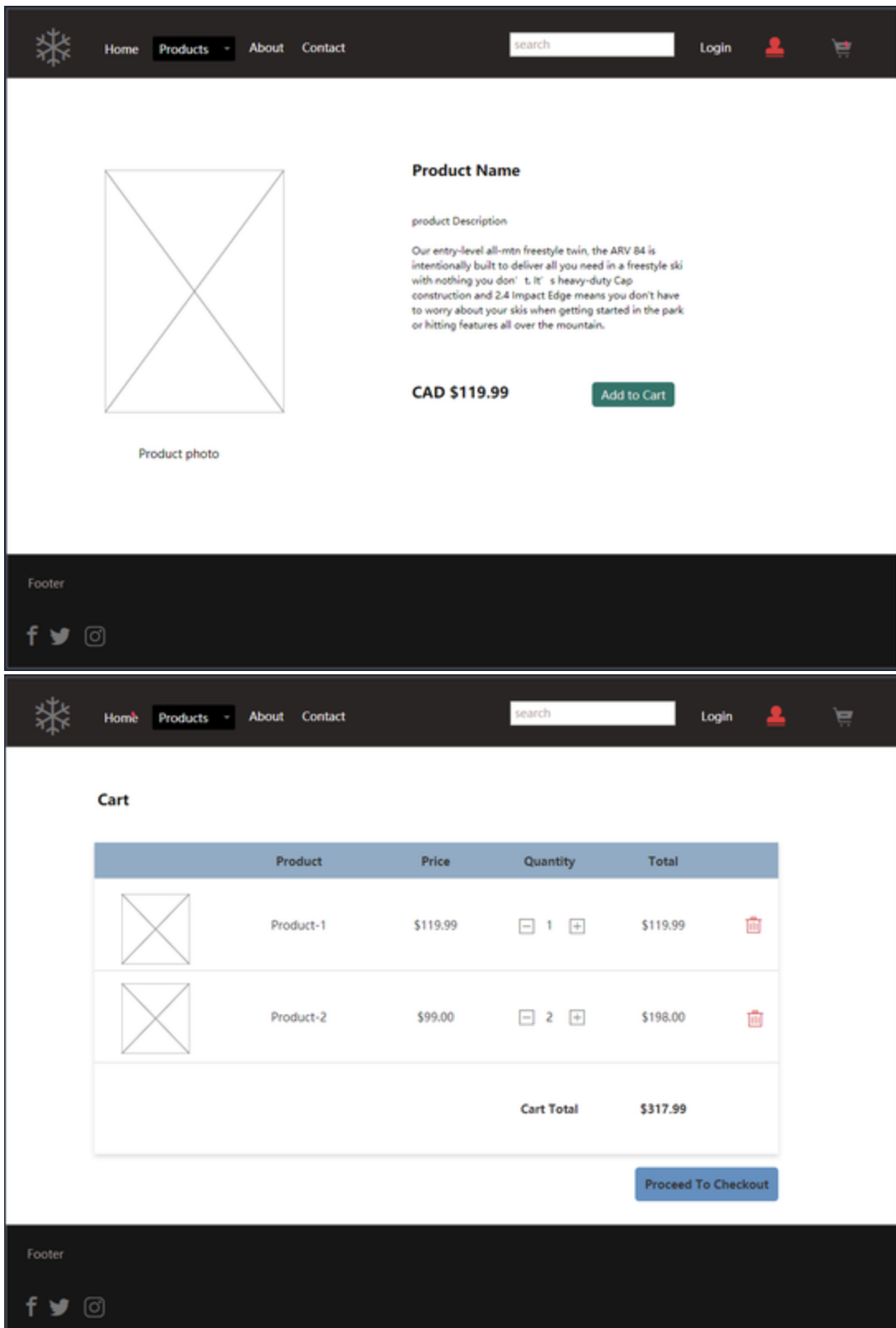
Initial database ERD, 6 tables at the beginning:

- users
- equipment
- reentrates
- orders
- cartItems
- orderItems



Mockups/Screenshots of Website main pages:





Source Control:

[Bitbucket](#)

Project Management:

[Confluence](#)(issue tracking/meeting notes)

[Trello](#) (Tasks Management)

Technologies MUST use:

- PHP Slim, MeekroDB, Twig, Monolog
- AJAX
- Other libraries, (use Packagist to find them) e.g.
- Facebook login, Google login (or universal login), Google maps, ReCaptcha
facebook/graph-sdk or hybridauth/hybridauth
- jQuery libraries, e.g. menus, mobile devices support
- Bootstrap
- CSS for printing (media queries)
- multiple languages support (multiple templates in subdirectories 'en' 'fr') or gettext, maybe boronczyk/localization-middleware (search for solutions online) <https://twig-extensions.readthedocs.io/en/latest/i18n.html>
- HTML 5 canvas, camera support, GPS/location services
- charts in HTML5 usually
- calendar library, e.g. <https://fullcalendar.io/demos>