IERG4210 Web Programming and Security (2023 Spring)
Assignment Marking Guidelines

Revision history

v1.0 Modified on 23/03/2023
v1.1 Modified on 29/03/2023

General Guidelines

The assignment is designed to let students practice what they have learned in the course. Students must be aware of web application security throughout web development. The whole assignment is split into 6 phases, leading all the way to a creative and functional shopping cart upon completion. Students should take a real-world website, parknshop.com, as a reference. In the assignment, students are expected to understand and apply proper security design principles and programming skills, regardless of which programming languages and libraries the students desire to use. The marking checklist included in the next page therefore outlines only the general requirements with a result-oriented basis in order to encourage students' creativity. For detailed guidance, students should refer to both lecture and tutorial notes.

Submission Policy

Students are required to package all of their source code, a README file, and any external resources (e.g. database, images, css and js files) into a zip file and submit it to the Blackboard. Each phase is associated with a firm submission deadline.

- Late Submission Penalty -- Late submission will lead to your mark reduction by the formula 0.9^n, where n is the round-up number of days delayed (e.g., assume your score is S and your submission is 9 hrs late → $0.9 \times S$, 25 hrs late → $0.81 \times S$, 49 hrs late →$0.729 \times S$, and so forth).
- *Final Demonstration* – Students will sign up for a timeslot to demonstrate their websites to a marker, who will then grade it according to the checklist. The marker will then evaluate the student's understanding with questions.

Honesty in Academic Work

CUHK places very high importance on honesty in academic work submitted by students, and adopts a policy of *zero tolerance* on cheating in examinations and plagiarism. Students are NOT allowed to submit anything that is plagiarised. Therefore, we treat every assignment our students submit as original except for source material explicitly acknowledged. We trust that students acknowledge and are aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty.

IERG4210 Web Programming and Security (2023 Spring)
Assignment Marking Checklist v1

Phase 5: Secure Checkout Flow (Deadline: Apr 9 2023) (Subtotal: 15')

This is a tough phase, yet the most critical one, to escalate your website (building skill) to the next (professional) level. (You'll likely be offered a job if you can demonstrate such a level of web programming skills.) The implementation has already been outlined as below. Be prepared to spend a substantial amount of time debugging.

1. Sign up at https://developer.paypal.com/dashboard/ : _____ / 2'
    a. Create two sandbox accounts – a merchant account and a buyer account:
        i. A merchant account – save email and password in secret.json
        ii. A buyer account – save email and password in secret.json
    b. Create a sandbox application:
        i. an application – save the client ID and in secret.json
2. Create a submit button via PayPal standard checkout APIs: _____ / 1'
    a. Embed a Paypal Smart Button inside your shopping cart (check payment.php)
        i. Include the PayPal JavaScript SDK
        ii. Set up a container element for the button
        iii. Render the button by *paypal.Buttons().render()*
3. When the checkout button is clicked, *createOrder()* in payment.php is called: _____ / 4'
    a. payment.php passes the *pid* and *quantity* of every individual product (or any other data) to create_order.php.
    b. payment.php waits for create_order.php to call *create_order()* to generate a well-formed order JSON string:
        i. create_order.php calls *gen_digest()* to generate a digest that is composed of at least:
            1. the *pid* and *quantity* of each selected product,
            2. The current price of each selected product gathered from DB,
            3. The total price of all selected products,
            4. Currency,
            5. Merchant's email address, and
            6. A random salt
        ii. create_order.php calls *gen_uuid()* to generate an invoice, and set it in a field *invoice_id*.
        iii. create_order.php puts the generated digest and invoice into the custom_id and invoice_id of an order.
        iv. create_order.php generates other necessary fields of an order. See https://developer.paypal.com/docs/api/orders/v2/ .
    c. payment.php submits the order now to PayPal using the *actions.order.create()* function.
4. After the buyer has completed the payment, *onApprove()* in payment.php is called: _____ / 2'
    a. payment.php passes the order details from PayPal to save_order.php.
        i. save_order.php saves the order into the DB.
    b. payment.php calls *clearCart()* to clear the shopping cart
5. After the buyer has finished paying with PayPal, auto-redirect the buyer back to your shop _____ / 1'
6. Display the DB *orders* table in the admin panel: product list, payment status…etc. _____ / 2'
7. Let members check what they have purchased in the most recent five orders. _____ / 3'
    a. Show the order information in the member portal.

References:
https://developer.paypal.com/docs/checkout/standard/integrate/
https://developer.paypal.com/sdk/js/reference/