Phase 6: Extensions (Deadline: Follow Final Report Deadline) (Subtotal: 9', Bonus: 7' max)

In this phase, you can choose any combinations of the following items to implement. At most 7' bonus will be awarded.

1. Mashup: Including a social plugin in the main page ________/ 2'
   - Facebook: https://developers.facebook.com/docs/plugins/
2. SEO: Apply search engine optimized (or user-friendly) URLs when browsing products ________/ 4'
   - Include the name of categories and products into the URLs:
     e.g. https://yourdomain.com/2-Fruits/ for browsing products under the category Fruits
     e.g. https:// yourdomain.com /2-Fruits/9-Apple for browsing product details
   - You can map the above URLs to your Node using Apache scripts (Hint: google RedirectCond)
   - Or handle it using Node param in path
   - (Note) Full bonus will be obtained only if SEO can be automatically applied to all products, including ones newly inserted.
3. Supporting pagination/AJAX infinite scroll when browsing products in the main page ________/ 6'
4. Supporting HTML5 Drag-and-drop file selection in the admin panel ________/ 4'
   - Create a dropping area that takes an image
   - Display a thumbnail (i.e. smaller width and height) if the dropped file is an image; reject it otherwise
5. Supporting multi-session management ________/ 6'
   - Show the simultaneous logged-in sessions in the admin panel
   - Each session should be identified by an IP and allows logging out other sessions
   - Hints: Use DB to save valid authentication token. Examples: Gmail and Dropbox
6. Supporting the use of gift vouchers (e.g. EASTER12 for $5 discount) ________/ 8'
   - Create a DB table called vouchers that store voucher code and the corresponding discount
   - Add a field for voucher code just above the checkout button
     - Auto fill the voucher code if it is supplied through a query parameter (e.g. ?vcode=EASTER12)
   - Use AJAX to dynamically validate the coupon code (using onkeydown/onblur handler)
   - Apply discount and update the UI to reflect the discounted price and the discount amount
   - Security: Make proper validations throughout the checkout process
   - Hints: HTML variables in p. 433 of the first reference
7. Supporting Secure Authentication with OAuth or WebAuthn ________/ 9'
   - Google: http://code.google.com/apis/accounts/docs/OAuth2.html
   - Facebook: https://developers.facebook.com/docs/authentication/
   - WebAuthn: https://webauthn.io/
8. Supporting secure password reset through email ________/ 6'
   - A page that asks for email address for password recovery
   - Only if the email corresponds to an existing user, an email will be generated
   - In the email, a password recovery hyperlink will make use of a random nonce
   - Only the admin receiving the nonce can reset his/her password
9. Supporting member management for buyers ________/ 6'
   - Create a member portal for buyers – sign up, sign in,sign out and change password
   - Let members check what they have purchased in the most recent N orders
10. Supporting discounts when purchasing multiple quantities of a product type ________/ 9'
    - Create a DB table called discounts that store conditions for applying discounts
      - Conditions could include: “buy 2 get 1” and “buy $10@2, $6@1”
      - Refer to parknshop.com for reference and details
    - Update the UI to reflect the discounted price whenever the quantity conditions are met
    - Security: Make proper validations throughout the checkout process

11. Apart from Paypal/Stripe, there are many other payment approaches, like Alipay, WeChat Pay and so on. In this step, you will add a second payment approach to your website ____/ 12'
12. Design an online chatbox on your website, users can click on it and send comments/enquiries to the customer service. _______/ 12'
    - Users' comments should be recorded and displayed in the admin panel.
    - The customer service should be **interactive**.
13. TBD – May release more options upon students' requests
    - Student can demonstrate it in the Final Report

PHASE 7: PEER HACKING (1 May to 4 May) (SUBTOTAL: 15', BONUS: 10' MAX)

It is critical to defend against potential attacks *before* they turn into reality when the website is open to the public. Students are to exercise *ethical hacking* in this phase. Be reminded to backup everything (code, conf files).

1. Practice with the use of any automated vulnerability scanner ________/ 0'
   - Use an automated vulnerability scanner (e.g., Nikto, Skipfish or others). Fix your vulnerabilities, if any.
   - **Scan ONLY your own website** to see what common errors you could have made
   - Note: Beware of exceeding the bandwidth quota so that you are charged

IF your website does not allow user registration, leave a visible user account with password at the FRONT PAGE

Can be reported as (Bug)

This is a game to help you learn security practically, which starts on 1 May to 4 May. Site owners should responsibly react to the incidents and fix the problems as soon as possible to mitigate future exploits. Each student is entitled a base score of 15, while the highest score is 25. Students must report any issues **at the Google Form with a Unique ID (posted in Blackboard)**:

1. Perform manual ethical hacking for shops of your classmates ________/ 15'
   - For a vulnerability in Student V's shop being reported by Student R1: R1 + 4'
     - Student R1 reports and clearly specifies the problems: V - 2'
     - **Only the first 5 entries a student submits will be counted**. We will sort your records based on the time you submit. Cherish your chances and make sure the bug really exists.
     - When reporting, Student R1 MUST strictly follow the following format
       - Shop Name with vulnerability: e.g. s81
       - Your SID/shop ID
       - Type of Vulnerability: Prefixed with OWASP Vulnerability types, e.g. [XSS]
       - Content: Answer the Questions in the report link
       - Attach at least one screen capture(s) for a proof and to aid illustration
     - If a bug report is invalid:
       - To dispute a bug, student V must load his/her source code from eLearn and demonstrate it during Video Demo
     - Student V fixes the problem on or before Submitting the Video Demo
       - Should point out the solutions in the demo
       - **No marks** will be deducted
   - For a non-security related bug in Student V's shop being reported by Student R:

     Please understand that some students might skip doing a part or two, regarding them as bugs to report are meaningless to the aim of this phase. So, unless your discovery is a kind of system flaw, otherwise, do not report it.
     - The marks allocation is half of the security ones: R1 + 1'
     - Each bug MUST be prefixed by the numbering of assignment requirements V -1'
       - Example number: [P4-1] stands for requirement 1 in phase 4
     - The bug must be reproducible in both Firefox and Chrome

     Each student can report **at most 3 non-security related bugs**
   - Availability checks by the automated bot
     - https://s00.ierg4210.ie.cuhk.edu.hk/status/shops will report the availability of shops during the whole period
     - **Deduct 2 mark to the OVERALL score if not functioning for 1 day:** V -2' / day
     - Also, do not hide your website page intentionally
   - Student R causes a high volume of traffic on others' websites
     - Includes but not limited to launching automated scan, DoS, or DDoS
     - Please shut the service to avoid being charged and submit the access_log to us ASAP

This game is expected to be quite exciting! :) This is an internal ethical hacking exercise; hence, students whose websites are exploited or found vulnerable should not take it as offensive but as a good learning opportunity. Students should respect each other and be polite in any circumstances. In case of dispute, TA will be the final judge on the marking. Drastic circumstances (e.g. bullying) can result in penalties when the instructor deems it appropriate

Final Report (Deadline: 9 May)

Students need to document the completion of all the **(key) features** required. Here is the suggested rundown.

1. Phase 1. Showing your AWS/Azure/Cloud services dashboard that you have set up the VM, virtual networks, and firewalls correctly.
   A. Show that your website is accessible via the URL.
   B. Show that the response header does not show the server language/version, and the direction index is disabled.

2. Phase 2.
   A. Show your main page with products and categories.
      i. Display products under a specific category
      ii. Products will have their detailed page.
   B. Admin can create/delete products and categories.
      i. Automatic image resizing is performed. (Add a product with a large image, and show its detailed page)

3. Phase 3. Shopping Cart is updated without a page load
   A. Add products to the shopping cart, and hover over the cart to show the added item
   B. Add quantity of a product, and the total price is updated without page load.
   C. Decrease the quantity of a product to 0 and it should be removed from the cart

4. Phase 4. Authentication for Admin Panel
   A. Show that your admin panel-related paths are authorized users only. Try to access it without login/user login/admin login. (also demonstrate the logout function)
   B. Show your XSS, CSRF, SQLi, and other defense mechanism.
   C. Show that the user can change the password

5. Phase 5. Checkout Workflow
   A. Perform a checkout on the products in the shopping cart.
   B. Show that the admin can track orders in the admin panel. (Created but not verified; You may open another browser in Incognito mode and act as user)
   C. An order can be verified after the user completes the payment.
   D. User can check his/her last order(s)

6. Phase 6. Any extension you have implemented.
   A. Explain what feature(s) you have added.
   B. High level idea of how it is implemented. You may show some of your source code

7. Any Security bugs fixed.

Final Q&A (if applicable) (Subtotal: -90')

Question forms may include but not limited to

- Illustrate a piece of code in your assignment, what does it do and why did you write like this
- Show how you protect your website from a certain type of attack
- Delete a few lines of your code and ask you to recover it

SID: ________________________________ Total: ______/ 122 (+19 bonus)