**Course code and name: COSC2430 - Web Programming**

**Assessment name: Full-stack Web Application**

**Length:** Code submission

**Type:** Group of 3 or 4 students

**Feedback mode:**Feedback will be sent by email/Canvas to students

**Late work:** A penalty of 10% (of the marks awarded) per day will apply; if more than 10 days late, a penalty of 100% will apply; weekend days (Saturday and Sunday) are counted when counting total late days

**Learning Objectives Assessed**

* **CLO1:** Understand modern languages, constructs used on the Web (such as HTML5, Advanced CSS, JavaScript (frameworks), and PHP)
* **CLO 2**: Explain the functions of clients and servers on the Web, and describe the strengths and weaknesses of the client-server internet approaches to web design and implementation
* **CLO3:** Program, access, and manipulate data through the adoption of accepted standards, mark-up languages, client-side programming, and server-side programming
* **CLO4:** Design and implement an interactive web site(s) with regard to issues of usability, accessibility, and internationalization
* **CLO5:** Design and implement a client-server internet application that accommodates specific requirements and constraints, based on analysis, modelling or requirements specification

**Ready for Life and Work**

Prepare students to work on software development projects that use web technologies

**Assessment Details**

**1. Instructions**

Students are asked to work **in groups** for this assignment and are encouraged to seek help from classmates and teaching staff. Each group needs to create a new GitHub repo and collaborate on that central system. All source code and resources must be present on GitHub too. Moreover, all commit messages must be meaningful and consistent.

Each group has to create a short video (5 to 10 minutes) to present the final result. The video must show all functions that have been completed. The video must be uploaded to YouTube or OneDrive. The link to the video must be included in your submission's README file (described later). NOTE: do NOT submit your video to Canvas, it is too big to receive directly.

**Note:**Your Canvas submissions will be used for grading.

**2. Background**

After completing the RMIT's Web Programming course with an HD grade, you feel very motivated and decide to found a start-up specializing in mobile and web application development. Despite its small size and lack of experience, your company aims to compete directly with larger firms by providing innovative design, enjoyable UI/UX, and timely support services. Now, the only thing you need is to have very challenging projects to prove your start-up's worth.

By actively participating in networking events and knowledge-sharing workshops, you've increased significantly the number of people who can recognize your business name and logo. Founded only one month ago, your start-up has successfully built and launched more than 10 small and medium websites. You feel OK but not so exciting. You want to do something that can scare you!

Eventually, you get one today.

Your client is a multimillionaire and world-class serial entrepreneur. She has created and sold businesses in diverse industries such as education, healthcare, and event management. Now, she wants to launch a web service similar to Instagram. The name for this new service is **InstaKilogram**. At this stage, to test the market, she asks your team to help develop the service MVP (Minimum viable product).

Seize this opportunity to make your name unforgettable.

**3. User Roles**

There are 3 main user roles in InstaKilogram:

**Admin:** admins can do everything on the website, for example, they can manage users and their uploaded images.

**User:** users are people who have accounts in the InstaKilogram system and have logged in when browsing the site.

**Guest:** guests are people who have not logged in when using the site (they may or may not have accounts in the system).

**4. UI Requirements**

The website must support the following device types

* Desktop: 1024 pixels or higher width
* Tablet: 768 to 1023 pixels width
* Smartphone: 767 pixels or lower width

The following sections are common on every page of the website

* Header: contains a logo, the website name, a search field, and "My Account" or "Login" link depending on whether the current user has logged in or not
* Footer: contains About, Copyright, Privacy, and Help links
* Main: this section is positioned between Header and Footer. Its actual content changes from page to page

For static pages (for example, About, Privacy, etc.): you can enter any meaningful information. You can copy from the Internet too, but you need to cite the source.

**5. Functional Requirements**

5.1. Register Account

When a visitor clicks on the Login link, display a login form (described later) and a "Don't have an account? Register a new one" link. If the register link is clicked, display a registration form. The following fields are required to register an account:

* Email address (must be unique for all users)
* Password
* Retype password
* Profile picture (an upload file field)
* First name
* Last name

There are 2 buttons on the Register form

* Clear: reset button
* Register: submit button

You need to validate user-entered data at both client-side (using Javascript) and server-side (using PHP). Use the following rules for your validation

* The email address must be in a proper format
* There is no duplication of email (you need to validate this condition at the server-side only). Note that email addresses are case-insensitive (i.e., abc@gmail.com is the same as AbC@GmAiL.com)
* Password must be from 8 to 20 characters. Each password must contain at least 1 lower case letter, at least 1 upper case letter, at least 1 digit
* Password and Retype password must be the same (validate at the client-side only)
* Profile picture: only files with the following extensions are allowed: jpg/jpeg, png, and gif (validate at the server-side only)
* First name and last name: any values from 2 to 20 characters

If registration is successful, the user information is stored on a file accounts.db on the server (Note: NO database management system can be used). You must put the accounts.db file outside of the document root to prevent it from being downloaded by hackers. The passwords of users must be hashed before storing to ensure their confidentiality. Refer to [PHP password hashing (Links to an external site.)](https://www.php.net/manual/en/function.password-hash.php) for more information.

5.2 Login/Logout and My Account

To log in, a correct combination of email and password is required.

The My Account page displays all information of the logged-in user, except the password. Note that a user must log in successfully before he/she can access this page. If a user has not logged in, but entered the My Account page URL directly into the browser address bar, that user will be redirected automatically to the Login page.

On the My Account page, the user can change her profile image to a new picture.

On the My Account page, there is a link to let the user log out of the system.

5.3. View Images

The main section (between the header and footer) contains 2 parts: a small subsection at the top that allows users to share images (this part is only visible to logged-in users) and a large subsection that displays images the users can view. The table below shows which images a user can view (sharing level is explained later shortly)

|  |  |
| --- | --- |
| Guests | Can see images that are shared publicly |
| Logged-in users | Can see images that are shared publicly and internally; plus images that are shared by this logged-in user |

The images are displayed according to their shared time, and the most recent images are displayed first. When an image is displayed, the description of the image (explained later) is displayed too.

5.4. Upload and Share Images

After logging in, a user can upload and share images. At the moment, only one image can be shared at a time.

The top of the main section contains a form for logged-in users to upload and share their images. To share an image, users choose an image file, enter a description (use the HTML text-area element) and choose a sharing level. Sharing level is a combo box with 3 options

* Public: anyone (including guests) can view
* Internal: only logged-in users can view
* Private: only the user who shares can view

5.5. GDPR Compliance

This website must comply with GDPR ([General Data Protection Regulation (Links to an external site.)](https://gdpr-info.eu/)). On every page, display a cookie consent message (see the attached image below) to visitors. If a visitor has agreed already, this message will not be displayed again on any pages. This function must be implemented using Javascript.

5.6. User and Image Management

The admins of InstaKilogram can manage user accounts and images. For now, you can create a hard-coded account for the admins. Furthermore, there is a separate login page for the admin (you don't need to link to this login page from the website - in other words, the admin can enter the login URL directly to access it).

After logging in, the admin can do the following

* View a list of all user accounts (pagination is required due to a possibly large number of accounts). Use the registration time to sort the accounts. The most recent registered account is displayed first
* Search user accounts based on email and name. If the search text is "ABC", any accounts whose emails or first name or last name contain "ABC" (case insensitive) must return. For example, "abcD" and "AbC" both contain "ABC". But "ABDC" does not contain "ABC". If there are many matching accounts, display them based on registered time in which the most recent account is displayed first
* In the user list or user search result, clicking on a user account will open a page to view the detailed information of that user. On this detailed page, the admin can reset the password of the user too
* View a list of images (admin can view all images, even private images). Most recent images appear first
* Delete an inappropriate image

**6. Non-functional Requirements**

You must use HTML elements, especially semantic elements, to structure the website in a meaningful way

Use external CSS for the styling. Do NOT use HTML to style the website

I will use the following sites to validate your HTML and CSS code

[https://validator.w3.org/ (Links to an external site.)](https://validator.w3.org/)

and

[https://jigsaw.w3.org/css-validator/ (Links to an external site.)](https://jigsaw.w3.org/css-validator/)

The result of those validators are used for marking (see the rubrics below)

**Submission**

* Except for Bootstrap, no external HTML/CSS/Javascript/PHP libraries can be used
* You can use the following external resources: fonts, icons, images, audio, videos as long as all of the following conditions are satisfied:
  + You cite the external resources properly
  + The original authors allow you to use their works (either explicitly or implicitly)
  + Those resources are not commercial ones (that means you do not have to pay to get/use those resources)
  + I can access those resources at the time of marking
* You must create a short video (5 to 10 minutes) demonstrating all features of your website (the ones that you've finished)
* The following files are required: HTML, CSS, JavaScript, PHP, and README.txt
* Directly inside your submission, there is a folder named "www" (without quote). Inside the www folder, there must be an "index.php" file. This is the home page of your web application. I will start your application from this index.php file
* A README.txt file is required in the root folder of your submission to describe briefly your group (must have all members' full names and IDs). If members' contributions are unequal, include in the README file the contributions (as percentages) of members. All members must discuss and agree on the contribution percentages. The total percentage must be 100%
* Contribution score:
  + If all members contribute equally: each earns 5 points
  + If a member contributes zero percent: this member earns zero for the WHOLE assignment
  + If members contribute differently: call X the percent value if all members contribute equally (i.e., X = 25 if a group has 4 members; X = 33 if a group has 3 members, etc). Who contributes <= 0.5X earns 1 point; who contributes > 0.5X but <= 0.75X earns 3 point; who contributes > 0.75X but <= 1.25X earns 5 points; who contributes > 1.25 earns 7 points (but if your total score is > 45, you still get just 45)
  + You can see that the above rules are not balanced between penalty and reward. I deliberately decided so to encourage you to contribute as equally as possible
* The README.txt file also contains the GitHub URL of your repo. You need to add me as a collaborator to your repo. My GitHub ID is TriDang
* The URL to your demonstration video must be included in the README file. Note: your video must be uploaded to YouTube or OneDrive
* You must explain how to start/run your website in the README.txt file
* You must create several user accounts, upload some images so that I can test the work more easily
* The emails, passwords of existing users and the admin must be included in the README.txt file for me to test
* Compress all files as one .zip file and submit the .zip file to Canvas. You need to ensure that I can uncompress your submission with popular software like 7-Zip. If I can't open your file or your file is broken, it is considered that you haven't submitted the assignment

**Support Resources**

Additional library and learning resources are available to help with the assessment in this course

* Library services
* Program Tutors
* Learning Advisors

**Rubric**

**Full-stack Web Application - Se**