1st SIT COURSEWORK 02 QUESTION PAPER **Year Long 2023 2024**

Module Code:	**CS6004NI**
Module Title:	**Application Development**
Module Leader:	**Mr. Bishal GC** (Islington College)

Coursework Type:	**Groupwork**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**Friday, 3 May 2024, before 01:00 PM**
When Coursework is given out:	**Week 22**
Submission Instructions:	Submit the following to the Islington College's MST portal before the due date: ● The software application to be developed in C# ASP.NET ● The documentation in PDF format by the **Group Leader only**
Warning:	London Metropolitan University and Islington College take Plagiarism seriously. Offenders will be dealt with sternly.

Plagiarism Notice

You are reminded that there exist regulations concerning plagiarism.

Extracts from University Regulations on Cheating, Plagiarism and Collusion

Section 2.3: “The following broad types of offence can be identified and are provided as indicative examples ….

(i) Cheating: including copying coursework.
(ii) Falsifying data in experimental results.
(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate’s assessment.
(v) Collusion to present joint work as the work solely of one individual.
(vi) Plagiarism, where the work or ideas of another are presented as the candidate’s own.
(vii) Other conduct calculated to secure an advantage on assessment.
(viii) Assisting in any of the above.

Some notes on what this means for students:

(i) Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.
(ii) Taking extracts from published sources without attribution is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. " e = mc2 (Einstein 1905)". A reference section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Further information in relation to the existing London Metropolitan University regulations concerning plagiarism can be obtained from http://www.londonmet.ac.uk/academic-regulations

Software Development Scenario

Bislerium PVT. LTD. is seeking a web application for **blogging**, which will eventually be integrated into their social media platform. Your group has successfully secured the initial prototype project through an auction. The project comes with stringent specifications, mandating the use of an enterprise-level framework.

Initially, the project has two categories of users: **bloggers and admin(s)**. Both bloggers and admin(s) are authorised personnel granted access to the application's features. Additionally, a user category referred to as "**surfers**" exists, allowing them to only browse through blogs and read them **without the need to log in**. Blog must consist of various sections, including but not limited to a **title, body, and images**.

Users must have the capability to self-register and log in to **post**, **react**, and **comment** blogs. The blogs can be **paginated** and **sorted** by **random**, **popularity** and **recency**. Each blog can then be subjected to community-driven **reactions (i.e., upvote, downvote) and comments.** Comments can be reacted to too. Authors of the blogs should receive **push notifications** upon the activities accumulated by their posts.

Admins, on the other hand, are exclusively provided with an overview of the system's usage. The **visualisation** encompasses various metrics, including the **all-time** count of blog posts, upvotes, downvotes, and comments, as well as the daily count of these activities within a **chosen monthly time frame**. It also includes **ranking** of the **top 10 most popular posts and bloggers** of all time and chosen month. To calculate the popularity of a blog post, a simple formula is used:

*Blog popularity = upvote weightage * upvotes + downvote weightage * downvotes + comment weightage * comments*

The weightage values can be customised based on preferences, with the standard weightage rates set as follows:

Activity	**Weightage**
Upvote	2
Downvote	-1
Comment	1

In terms of **individual popularity**, it is determined by the **cumulative sum** of popularity scores for all blog posts published by the respective author.

[We encourage students to leverage external/3rd party libraries.]

DELIVERABLES

Your submission should include the software project and a reflective essay as described below.

1. Your **software artefact** is in the form of a **Visual Studio 2022** or above project, which should include the program's **source code** and **data file** (if any). Make sure to exclude **.git**, **.vs**, **.github**, **bin**, and **obj** folder.

2. A **reflective essay** (1200 words), which concisely documents:

    a. detailed instructions to run the program.

    b. concise description of your solution design for each of the implemented functions of the application. Please do not include code listing here.

    c. the architecture of your software in terms of software classes and their purpose, clearly indicating which classes to be of your own work and which classes from other sources (e.g., from textbooks, online sources such as MSDN etc.)

    d. detailed description of the classes' properties and methods

    e. **individual** group member's reflection of own experience of using Visual Studio, C# ASP.NET Core and MS SQL to develop a web application interfacing a backend database, e.g., which feature you like and why, what issues you experienced and your solution to overcome it.

Marking Scheme

Section	Topic	Full Marks
A	**Implementation of Application**	**80**
1.	Users can register, view, update, and delete their profile.	6
2.	Users can change the password along with resetting via email.	4
3.	Bloggers can create, edit, and delete their own blog posts.	10
4.	Each image file attachment must not exceed greater than 3 Megabytes (MB).	4
5.	Users can surf through a paginated and sortable catalogue of blogs. The catalogue can be sorted by random, popularity, and recency.	6
6.	Anonymous users also called surfers can view and read blogs without needing to login.	4
7.	Bloggers have the option to upvote or downvote both their own and others' blog posts.	6

8.	Bloggers can comment in their own and others' blog posts. Comments can also be reacted to either upvote or downvote.	6
9.	Blog Comment can be updated or deleted by the author (one who commented).	4
10.	Every activity, such as reactions or comments, within a post should trigger a push notification to the respective author or blogger.	10
11.	The history of updates for both blogs and comments are preserved.	4
12.	Admin can create another admin.	4
13.	Admin has only access to a dashboard that visualises both all-time and month specific cumulative count of blog posts, upvotes, downvotes, and comments.	6
14.	The dashboard also includes the top 10 most popular blog posts and bloggers of all time and specific month.	6
B	**Documentation**	**10**
1.	Detailed instruction to run the program	2

2.	Concise description of your logical solution to each of the implemented functions of the application.	2
3.	The software architecture, i.e., class and their purpose	2
4.	Detailed description of the classes' properties and method	2
5.	Individual member's own reflection of own experience	2
C	**Programming Style**	**10**
1.	Clarity of code, Proper Naming conventions & comments	2
2.	Sensible naming of programmer-defined variables, classes, properties, and methods	2
3.	Useful comments in code	2
4.	Data validation and exception handling	2
5.	Interface design and usability of the system	2
	Total Marks	**100**

End of paper