

Module Code:	CS6004NT
Module Title:	Application Development
Module Leader:	Mr. Bikram Poudel (Islington College)

Coursework Type:	Individual
Coursework Weight:	This coursework accounts for 30% of the overall module grades.
Submission Date:	Week 14 (January 28, 2026)
Coursework given out:	Week 8
Deliverables:	<p>1. Documentation:</p> <ul style="list-style-type: none"> ○ Report (PDF format) <p>2. Development:</p> <ul style="list-style-type: none"> ○ C#.NET Project Repository (Zip file) ○ Published App (Optional) <p><i>Note: Exclude dot-folders (.git, .vs, .vscode) and build output folders (bin, obj) from project repository.</i></p>
Submission Instructions:	<p>Submit the deliverables to the Itahari International College MST portal before 2:00 PM on the due date. You have two submission options:</p> <ul style="list-style-type: none"> • Option 1: Upload the zip file containing all deliverables. • Option 2: If the first option is not feasible, submit only the report. Ensure the report includes a link to access the Development Deliverables. <p>File Naming Convention: <London met Id> <Name>.pdf zip> E.g. 22016032 Aryan Jha.zip</p>
Warning:	London Metropolitan University and Itahari International College take plagiarism very seriously. Offenders will be dealt with sternly.

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed overleaf. Please sign below to say that you have read and understand these extracts:

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 taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; 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:

1. 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.
2. 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. “ $e = mc^2$ (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.

CONTRACT CHEATING

Contract cheating (also known as assessment outsourcing, commissioning, or ghost writing) is when someone seeks out another party, or AI generator service, to produce work or buy an essay or assignment, either already written or specifically written for them or the assignment to submit as their own piece of work.

Contract cheating undermines the integrity of the academic process and devalues the qualifications awarded by the university. Students are reminded that academic integrity is a fundamental principle of our institution. Engaging in contract cheating not only impacts the individual's academic record but also the reputation of the university.

Students are encouraged to seek support if they are struggling with their coursework. The university offers a range of resources, including academic counselling, tutoring services, and workshops on study skills and time management. Utilizing these resources can help students achieve their academic goals without resorting to dishonest practices.

Penalty:

- Failure in the Module: The student must re-register for the same module, and the re-registered module will be capped at a bare pass.
- Ineligibility to Continue on the Course: Where re-registration of the same module, or a suitable alternative, is not permissible, the student will not be able to continue the course. Additionally, the following penalty will be applied to the student's final award:
 - Undergraduate Honors: The student's final classification will be reduced by one level.
 - Unclassified Bachelors: Downgraded to Diploma in Higher Education.
 - Foundation Degree: Distinction downgraded to Merit; Merit downgraded to Pass; Pass downgraded to Certificate in Higher Education.
 - Masters: Distinction downgraded to Merit; Merit downgraded to Pass; Pass downgraded to Postgraduate Diploma.

Reporting and Consequences:

Instances of contract cheating will be thoroughly investigated, and students found guilty will face the penalties outlined above. It is the responsibility of every student to ensure that their work is their own and to avoid situations that could lead to accusations of academic misconduct.

By adhering to these standards, students contribute to a fair and equitable academic environment, ensuring the value and recognition of their qualifications are maintained.

SCENARIO

A writer wants to modernize their personal journaling process with a **secure, feature-rich desktop application**. The system must allow the user to **create, update, and delete one journal entry per day**, supporting **rich-text or Markdown formatting** for content. Each entry is tied to a **date and time**, with system-generated timestamps for creation and updates.

Users should be able to **track their moods**, selecting **one primary mood** and up to **two secondary moods**, categorized as **Positive, Neutral, or Negative**. Additionally, users can **tag entries** with custom or pre-defined tags (e.g., Work, Health, Travel, Fitness) and organize entries under categories.

The system must provide easy navigation of past entries through a calendar view and a paginated timeline/list view. Users should be able to search entries by title or content and filter by date range, mood(s), or tags.

To encourage consistent journaling, the application will track **daily streaks, longest streaks, and missed days**. A **dashboard** will provide analytics, including mood distribution, most frequent moods, most used tags, tag breakdown, and word count trends over time.

The application must also support **security and privacy** via password or PIN protection, allow **exporting journals as PDF** by date range, and store all data **locally in an SQLite database**. Optional features include **theme customization (light/dark)** for a personalized interface.

Journal Structure

- **Rich-text/Markdown Content**
- **DateTime** → Created At, Updated At (System-generated)
- **Mood(s):**
 - **Primary Mood (Required):** One main mood for analytics
 - **Secondary Mood (Optional):** Up to two additional moods
 - Categories:
 -  **Positive:** Happy, Excited, Relaxed, Grateful, Confident
 -  **Neutral:** Calm, Thoughtful, Curious, Nostalgic, Bored
 -  **Negative:** Sad, Angry, Stressed, Lonely, Anxious
- **Category**
- **Tags (Optional):**
 - **Custom Tags**
 - **Pre-built Tags:** Work, Career, Studies, Family, Friends, Relationships, Health, Fitness, Personal Growth, Self-care, Hobbies, Travel, Nature, Finance, Spirituality, Birthday, Holiday, Vacation, Celebration, Exercise, Reading, Writing, Cooking, Meditation, Yoga, Music, Shopping, Parenting, Projects, Planning, Reflection
- **One Journal Entry per Day** → Supports daily streaks
- **CRUD Operations** → Create, update, or delete the entry for the current day
- **Search & Filter** → By content, date range, moods, or tags
- **Analytics (Date Range Filterable)**
 - **Mood Distribution** → Pie/Bar chart showing % of positive, neutral, negative moods
 - **Most Frequent Mood** → The most common mood recorded
 - **Daily Streak** → Current consecutive days of journaling
 - **Longest Streak** → Maximum streak achieved
 - **Missed Days** → Dates with no entries
 - **Most Used Tags** → Bar chart/word cloud of frequent tags
 - **Tag Breakdown** → % of entries per category (e.g., Work, Health, Travel)
 - **Word Count Trends** → Average words per entry over time

TASK

The deliverables are outlined on the first page. Review the provided scenario and analyze it to extract all relevant requirements. For reference, the fixed requirements are detailed in the marking scheme below.

You are tasked with developing a **C#.NET software application** based on the given scenario. Additionally, clarify your contribution according to the project type (i.e., individual or group).

After completing the application, ensure it is thoroughly documented. Refer to the marking scheme for documentation requirements and review the "Things to Remember" section for further guidance.

THINGS TO REMEMBER

1. Plagiarism:
 - a. Plagiarism is grounds for failure and applies to deliverables. All parties involved will be penalized.
 - b. Code-level plagiarism is prohibited. Always provide proper attribution for any borrowed code.
2. Documentation:
 - a. Development without accompanying documentation (and vice versa) is not allowed and will result in failure.
3. Group Projects:
 - a. Groups can have up to 6 members. Only the group leader is responsible for submission.
4. Frameworks:
 - a. Any framework under C#.NET is permitted.
5. Deployment:
 - a. Deploying or publishing the completed application is encouraged.
6. Architectural Patterns:
 - a. Following well-known architectural patterns is encouraged.
7. Non-Functional Requirements:
 - a. Addressing non-functional requirements such as security, performance, scalability, compatibility, and usability will enhance your project's evaluation.
8. Development Tools:
 - a. Using project templates, libraries, packages, or modules to simplify the development process is encouraged, provided you justify their use.
9. Version Control:
 - a. Use version control systems (e.g., Git) to manage your codebase. Regular commits and clear commit messages are important.
10. User Experience (UX):
 - a. Pay attention to the user interface (UI) and user experience (UX) design. An intuitive and user-friendly application will positively impact your evaluation.
11. Project Management:
 - a. Demonstrate effective project management practices, including setting milestones, tracking progress, and managing resources. This will be reflected in your project review.
12. References:
 - a. Always provide references for any external resources used, such as libraries, remote repositories, articles, forums, or Q&A sites.

"We welcome and encourage your creativity."

MARKING SCHEME

SN	Features	Total Marks (100)
A	Development	90
AA	Feature	60
1	Journal Entry Management – Create, update, delete daily entry (only one per day with system timestamps)	5
2	Rich Text/Markdown Writing – Formatting support (bold, italics, lists, headings, links)	5
3	Mood Tracking – Primary mood required, up to two secondary moods	5
4	Tagging System – Custom or pre-built tags to classify entries	5
5	Calendar Navigation – Navigate entries through a calendar view	5
6	Paginated Journal View – View entries in paginated form for easy navigation	5
7	Search & Filter – Search by title/content; filter by date, moods, or tags	5
8	Streak Tracking – Track daily streaks, longest streak, and missed days	5
9	Theme Customization – Switch between light/dark or custom themes	5
10	Dashboard Analytics & Insights – Mood distribution, frequent moods, most used tags, tag breakdown, word count trends	5
11	Security & Privacy – Password/PIN protection of journals	5
12	Export Journals – Export multiple entries as PDF by date range	5

AB	Quality	30
1	Code Readability (i.e. naming convention, comments, indentation, consistent formatting/styles, organization, error messages, logical flow, expressions)	5
2	Code Efficiency (Data structures, algorithms, optimizations, redundant computations)	5
3	Code Modularity (SoC, code reusability, SRP, abstraction, Dependency Injection (DI))	5
4	Error Handling (exception handling, input validation, logging, graceful degradation, error propagation)	5
5	Version Control (VCS (i.e. Git)) , meaningful commit messages, logical commit history, effective branching/merging, Tagging/Releases)	5
6	User Experience (UX) (Design, Usability, Responsiveness, Consistency)	5
B	Documentation	10
1	Project Overview (purpose, scope, objectives)	1
2	Features & Functionalities (key features, corresponding functionalities, visual guides)	4
3	Proof of Work (PoW) (project artifacts: designs, UML diagrams, VCS repository, milestone chart, testing etc.)	2
4	Individual Reflection(s) (roles & responsibilities, personal insights, challenges, learnings, growth, impact on future work, personal evaluation)	2
5	Conclusion (objective restatement, implications, recommendations, findings, limitations, future research & development, concluding statement)	1

MILESTONE 1 (Week 9 (December 21, 2025))

- 1) **Task 1:** Initialize private Git repository with an empty MAUI project.
- 2) **Task 2: Project Report**
 - a) Project Overview
 - b) UI Design (Wireframe)
 - c) Data/Entity Modelling (E.g. Mood, Entry, User etc.)
 - d) Technology Stack
 - i) **Framework** (E.g. MAUI Blazor Hybrid, WinForms, Xamarin etc.)
 - ii) **External Libraries** (E.g. Newtonsoft.Json, Bogus, MudBlazor etc.)
 - iii) **Persistence Mechanism:** (e.g., file handling using local database SQLite)
- 3) **Task 3:** Must complete at least 2 features.

MILESTONE 2 (Week 11 (Jan 4, 2025))

- 1) **Task 1:** Must complete at least 5 features.

MILESTONE 3 (Week 13 (Jan 18, 2025))

- 1) **Task 1:** Must complete at least 5 features.