INDIRAGANDHINATIONALOPENUNIVERSITY

Regional Centre Delhi-3

Format for Assignment Submission

For Term End Exam June/December- JUNE **(Year)** 2025

(Please read the instructions given below carefully before submitting assignments)

1. **Name of the Student** : Ashu Chauraiya
2. **Enrollment Number** : 2501321326
3. **Programme Code** : MCA_NEW
4. **Course Code** : MCS-213
   (Use this format course-wise separately)
5. **Study Centre Code** : LSC-38046
6. **Name of the Study Centre With complete address** : RAJDHANI COLLEGE RAJA GARDEN NEW DELHI
7. **Mobile Number** : 8448713694
8. **E-mail ID** : Intermezzobrilliance@gmail.com
9. **Details if this same assignment has been submitted anywhere else also** : No
10. **Above information is cross checked and it is correct: Yes/No** ✓

Date of Submission: 01-06-25

Ashu

(Signature of the student)

A. Important Instructions:-

1. Please do not send any assignment at any email of the Regional Centre, it will not be considered.
2. Please avoid duplicacy. Do not re-submit the same assignment anywhere else or by any other means.
3. About the mode of submission of assignments, pl wait for instructions from IGNOU Hqtrs. As soon as we shall come to know, we will share it with all.
4. Please do not use plastic covers. Use plain A4 size pages for assignments for uniformity and better management with this cover page format on each assignment.
5. Please write your name and enrollment no. at the bottom of each page of your assignment.
6. Please retain a photocopy set of assignment submitted with you for record(may be asked to submit at later stage) and also keep the assignment submission receipt in safe custody.
7. If assignment awards are not updated in your Grade Card **within next 09 months**, please write to us at rcdelhi3@ignou.ac.in giving your complete details and attaching the proof of assignment submission.
8. Assignment Question Paper can be downloaded from: https://webservices.ignou.ac.in/assignments/

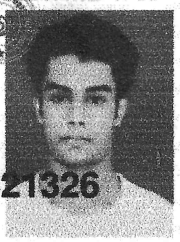
B. Compulsory sequence of the Assignment Set:

1. Duly Filled in Assignment Submission Cover Page (This Format Page).
2. Copy of IGNOU Identity Card.
3. Print out of valid/applicable assignment question paper.
4. Handwritten Assignment, written on both the sides of page (preferably plain A4 size).

ignou
THE PEOPLE'S
UNIVERSITY

इंदिरा गांधी राष्ट्रीय मुक्त विश्वविद्यालय
मैदान गढ़ी, नई दिल्ली - 110068
Indira Gandhi National Open University
Maidan Garhi,New Delhi - 110068

IGNOU - Student Identity Card



2501321326

Enrolment Number :	2501321326
RC Code :	38: DELHI 3 Naraina
Name of the Programme :	MCA_NEW : Master of Computer Applications
Name :	ASHU CHAURASIYA
Father's Name :	VIJAY
Address :	Flat No. 270,G, F Block-14, Pocket 13, Sector-20 RNORTH WEST
Pin Code :	110086

Instructions :
1. This card should be produced on demand at the Study Center, Examination Center or any other Establishment of IGNOU to use its facilities.
2. The facilities would be available only relating to the Programme/course for which the student is registered.
3. This ID Card is generated online. Students are advised to take a color print of this ID Card and get it laminated.
4. The student details can be cross checked with the QR Code at www.ignou.ac.in

Registrar
Student Registration Division

Course Code	:	**MCS-213**
Course Title	:	**Software Engineering**
Assignment Number	:	**MCA_NEW(I)/213/Assign/2025**
Maximum Marks	:	**100**
Weightage	:	**40%**
Last Dates for Submission	:	**30th April 2025 (for January Session)** **31st October 2025 (for July Session)**

This assignment has one question for 80 marks. 20 marks are for viva voce. You may use illustrations and diagrams to enhance the explanations. Please go through the guidelines regarding assignments given in the Programme Guide for the format of presentation.

Q1:

Assume that you are assigned responsibility of developing an **Online Date Sheet Generation System (ODSGS)** for a University. **ODSGS** should run both on PCs and Mobile Devices. **ODSGS** will have fields such as Course Code, Course Title, Date of Exam, Time of Exam etc. The user of ODSGS will input the list of course codes and course titles and duration of examination. Sunday will be holiday. Starting date of exams will also be input by the user. There may be other information that is needed by ODSGS to generate the date sheet. So, you can list them and make provision for them in database structure. Make assumptions wherever necessary. Based on the inputs received, ODSGS should generate the Date Sheet.
For developing **ODSGS** as specified above,

(a) Which SDLC paradigm will be selected. You may also suggest a SDLC paradigm that is proposed by you and non-existent as on date. Justify your answer. **(10 Marks)**
(b) List the functional and non-functional requirements. **(10 Marks)**
(c) Estimate the cost. **(15 Marks)**
(d) Estimate the efforts. **(15 Marks)**
(e) Develop SRS using IEEE format. **(15 Marks)**
(f) List queries for whom Reports can be generated. **(5 Marks)**
(g) List specific requirements which enables **ODSGS** to run on both PCs and Mobile Devices. **(10 Marks)**

Enrollment No. 2501321326

Question 1. (a)

Answer: For the Online Date Sheet Generation System (ODSGS), a suitable SDLC paradigm would be the Agile Model. This model is highly iterative and incremental, allowing for flexibility in requirements and continuous feedback from stakeholders, which is crucial for a system that needs to adapt to various university-specific needs and user inputs. Its iterative nature helps in developing and testing features in small cycles, ensuring that the system evolves effectively to meet both PC and mobile device requirements.

A non-existent SDLC paradigm I propose is the "Adaptive Intelligence-Driven Development (AIDD) Model". This model would integrate advanced artificial intelligence and machine learning throughout the entire SDLC. In AIDD, AI agents would continuously monitor development progress, analyze code for potential issues, predict future requirements based on user behavior patterns, and even suggest optimal architectural decisions. For ODSGS, AIDD could proactively identify potential scheduling conflicts, optimize exam timings based on student load and faculty availability, and automatically adapt the user interface for different devices based on real-time usage data, minimizing manual adjustments and enhancing efficiency.

Question 1. (b)

Answer: Functional requirements for ODSGS:

The system shall allow users to input course codes, course titles, and examination durations.

The system shall allow users to specify the starting date of examinations.

The system shall ensure that Sunday is automatically marked as a holiday and no exams are scheduled on this day.

The system shall generate a date sheet based on the provided inputs, including course details, exam dates, and times.

The system shall allow for the input of additional information necessary for date sheet generation, such as faculty availability, classroom capacity, and student enrollment numbers per course.

The system shall allow users to view, edit, and save generated date sheets.

The system shall provide an option to print or export the generated date sheet in various formats.

Non-functional requirements for ODSGS:

Performance: The system shall generate a date sheet within a reasonable time frame, even with a large number of courses. It should respond quickly to user inputs.

Usability: The user interface shall be intuitive and easy to navigate for all users, including those with limited technical expertise.

Reliability: The system shall be robust and available 24/7, with minimal downtime. It should handle errors gracefully and prevent data loss.

Security: The system shall ensure data integrity and confidentiality, with appropriate user authentication and authorization mechanisms to protect sensitive information.

Portability: The system shall be capable of running seamlessly on both

personal computers and mobile devices, adapting its interface and functionality accordingly.

Maintainability: The system's code base shall be well-documented and modular, allowing for easy updates, bug fixes, and feature enhancements.

Scalability: The system shall be able to handle an increasing number of users, courses, and data without significant degradation in performance.

Question 1. (c)

Answer: Estimating the cost for ODSUS development involves considering several key factors. Firstly, personnel costs will form a significant portion, covering salaries for software engineers, UI/UX designers, quality assurance testers, project managers, and business analysts. The number of personnel and their experience levels will directly impact this. Secondly, software and hardware costs include licenses for development tools, operating systems, databases (like a NoSQL database for flexible data structures), and potentially cloud infrastructure for hosting the application and database. Thirdly, training costs for the development team on new technologies or for end-users on how to operate the system will be incurred. Fourthly, testing and quality assurance costs involve resources dedicated to ensuring the system's functionality, performance, and security. Finally, miscellaneous costs such as communication, travel, and contingency for unforeseen issues should also be factored in. A common approach is to estimate effort first and then multiply by the average daily rate of the team members, adding fixed costs for software and hardware.

Question 1. (d)

Answer: Effort estimation for ODSGS development can be approached using several techniques. One common method is Function Point Analysis (FPA), where the system's functionality is broken down into measurable units (function points) based on user inputs, outputs, inquiries, internal logical files, and external interface files. Each function point is assigned a weight, and the total function points are then converted into effort (person-months) using historical data or industry benchmarks. Another technique is the COCOMO (Constructive Cost Model), which uses algorithmic cost estimation based on lines of code or function points, adjusted by various cost drivers like product attributes, hardware attributes, and personnel attributes. Alternatively, Expert Judgment can be used, where experienced developers and project managers provide estimates based on their past experience with similar projects. For ODSGS, the effort would be broken down into distinct phases: requirements gathering and analysis, system design and architecture, database design, front-end development (for both PC and mobile), back-end development, integration, testing, deployment, and documentation. Each phase would have its own estimated effort, summing up to the total project effort.

Question 1.

(e)

Answer: Here's an outline for the Software Requirements Specification (SRS) for ODSGS using the IEEE format:

1. Introduction

1.1 Purpose: This document specifies the functional and non-functional requirements for the Online Date Sheet Generation System (ODSGS).

1.2 Scope: The ODSGS will facilitate the automated generation of university examination date sheets, accommodating user-defined inputs and running on PC and mobile platforms.

1.3 Definitions, Acronyms, and Abbreviations: Define key terms like ODSGS, Course Code, etc.

1.4 References: List any documents referenced, such as university examination policies.

1.5 Overview: Briefly describe the rest of the SRS document.

2. Overall Description

2.1 Product Perspective: ODSGS is a standalone application that interacts with users for input and provides generated date sheets as output. It will not be part of a larger system initially.

2.2 Product Functions: Summarize the main functions: input course data, set exam parameters, generate date sheet, handle holidays, manage additional constraints.

2.3 User Characteristics: Describe the types of users (e.g., university administrators, examination department staff) and their skill levels.

2.4 General Constraints: Include any limitations such as budget, timeline, regulatory compliance, and existing university systems.

2.5 Assumptions and Dependencies: Assume availability of necessary

hardware/software infrastructure and cooperation from university departments for data provision.

3. Specific Requirements

3.1 External Interface Requirements:

3.1.1 User Interfaces: Describe the graphical user interface for PC and mobile, including screen layouts, navigation, and input fields.

3.1.2 Hardware Interfaces: Specify minimum hardware requirements for PC and mobile devices.

3.1.3 Software Interfaces: Detail any integrations with other university systems (e.g., student information system for course lists, if applicable)

.

3.1.4 Communications Interfaces: Specify network protocols and connectivity requirements.

3.2 Functional Requirements:

3.2.1 User Management:

Ability to authenticate users (e.g., login with credentials).

Role-based access control (e.g., administrator, data entry).

3.2.2 Course Data Management:

Input and validation of Course Code, Course Title, Examination Duration.

Ability to import course data from external files.

Ability to view, edit, and delete existing course records.

3.2.3 Exam Parameter Configuration:

Input of Examination Start Date.

Automatic identification and exclusion of Sundays as holidays.

Input of additional constraints (e.g., maximum exams per day, minimum gap between exams for a student, specific faculty unavailable dates, classroom capacities).

3.2.4 Date Sheet Generation:

Algorithm for automated date sheet generation based on all inputs and constraints.

Conflict detection and resolution mechanisms.

3.2.5 Date Sheet Viewing and Export:

Display generated date sheet in a clear, readable format.

Option to export date sheet as PDF, Excel, or other common formats.

Option to print the date sheet.

3.3 Performance Requirements:

Specify response times for key operations (e.g., date sheet generation within X minutes for Y courses).

Define concurrent user support.

3.4 Design Constraints:

Use of specific programming languages or frameworks.

Database system to be used (e.g., SQL or NoSQL).

3.5 System Attributes:

3.5.1 Security: Data encryption, access control, audit trails.

3.5.2 Reliability: Uptime percentage, error handling.

3.5.3 Maintainability: Modularity, coding standards.

3.5.4 Portability: Compatibility with various operating systems and browsers

3.5.5 Usability: User-friendliness, accessibility features.

4. Supporting Information

4.1 Appendix A: Data Model: Provide a high-level description of the database structure, including tables for Course, Exam, Faculty, Classroom, etc, and their relationships.

4.2 Appendix B: Glossary: Define any remaining technical terms.

Question 1. (f)

Answer: Reports can be generated from ODSUS based on various queries to provide useful insights and summaries.

Query 1: List all exams scheduled on a specific date, including Course Code, Course Title, and Time of Exam.

Query 2: Show the complete date sheet for a particular department or faculty, including all courses offered by them, their exam dates, and times.

Query 3: Identify any potential conflicts or overlaps in the generated date sheet that might require manual review, for example, two exams for the same student scheduled at the same time.

Query 4: Display a summary of exam durations for all courses, perhaps grouped by course type or level.

Query 5: List all courses that do not yet have an assigned exam date or time.

Query 6: Provide a report on classroom utilization, showing which rooms

are booked for exams on specific dates and times.

Query 7: Generate a list of all courses and their respective examination durations, sorted by course code or title.

Question 1. (g)

Answer: Specific requirements enabling ODSUS to run on both PCs and Mobile Devices focus on adaptability and user experience across different platforms

.

Responsive Design: The user interface must automatically adjust its layout, elements, and content presentation to fit various screen sizes and orientations, from large desktop monitors to small mobile phone screens. This ensures optimal readability and interaction without horizontal scrolling.

Touch-Friendly Interactions: All interactive elements, such as buttons, input fields, and navigation menus, should be large enough and spaced adequately to be easily tappable on touchscreens. The system should support standard touch gestures like tapping, swiping, and pinching.

Optimized Performance for Mobile Networks: The application should be designed to perform efficiently even on slower mobile data connections, minimizing data transfer and optimizing image and resource loading.

Platform-Specific UI/UX Considerations: While maintaining a consistent core experience, the design should subtly align with common UI patterns of desktop operating systems and mobile platforms (Android/iOS) to feel natural to users on each device.

Input Method Adaptability: The system should seamlessly switch between

keyboard and mouse input on PCs and virtual keyboard and touch input onmobile devices, ensuring all functionalities are accessible regardless of theinput method.

Offline Capability (Optional but beneficial): For mobile devices, a limited offline mode for viewing already generated date sheets could enhance usability in areas with poor connectivity.

Accessibility Features: Ensure the system is accessible to users with disabilities on both platforms, adhering to accessibility guidelines for screen readers, color contrast, and keyboard navigation.

Cross-Browser Compatibility: The web-based system must function correctly and consistently across various web browsers commonly used on both PCs (e g, Chrome, Firefox, Edge) and mobile devices (e g, Safari, Chrome for Android).