



# Model Curriculum

**QP Name: Junior Software Developer**

**QP Code: SSC/Q0508**

**Version: 4.0**

**NSQF Level: 4.0**

**Model Curriculum Version: 4.0**

IT-ITes SSC NASSCOM || NASSCOM Plot No - 7, 8, 9 & 10, 3rd Floor, Sector 126,  
Noida, Uttar Pradesh - 201303 || Email: standards@nasscom.in

# Table of Contents

## Contents

Training Parameters.....	3
Program Overview .....	4
Training Outcomes.....	4
Compulsory Modules.....	4
Module 1: Introduction to IT-ITeS Industry and Job Role of a Junior Software Developer .....	6
Module 2: Software Basics and Development Lifecycle .....	7
Module 3: Problem-Solving with Advanced Coding Techniques .....	8
Module 4: Database Management and Version Control .....	9
Module 5: Agile Development and Collaboration .....	10
Module 6: Ensuring Code Quality and Deployment .....	11
Module 7: Security Practices and Generative AI Integration.....	12
Module 8: Employability Skills (30 Hours) .....	13
Module 9: On-the-Job Training.....	15
Annexure.....	16
Trainer Requirements .....	16
Assessor Requirements.....	17
Assessment Strategy.....	18
Glossary.....	20
Acronyms and Abbreviations.....	21

## Training Parameters

<b>Sector</b>	IT-ITeS
<b>Sub-Sector</b>	IT Services
<b>Occupation</b>	Application Development
<b>Country</b>	India
<b>NSQF Level</b>	4
<b>Aligned to NCO/ISCO/ISIC Code</b>	NCO-2015/2512.0205
<b>Minimum Educational Qualification and Experience</b>	<p>12th Grade* Pass</p> <p>OR</p> <p>10th Grade Pass with 3-years of relevant experience**</p> <p>OR</p> <p>Previous relevant Qualification of NSQF Level 3 with 3 years of relevant experience**</p> <p>*With computer knowledge</p> <p>**Relevant field: Computer Operation</p> <p>The relevant experience would include work, internship and apprenticeship after completion of relevant educational qualifications.</p>
<b>Pre-Requisite License or Training</b>	Software Development Certifications in C++, Embedded, C#, C, Java etc.
<b>Minimum Job Entry Age</b>	15 years
<b>Last Reviewed On</b>	18-02-2025
<b>Next Review Date</b>	18-02-2028
<b>NSQC Approval Date</b>	18-02-2025
<b>QP Version</b>	4.0
<b>Model Curriculum Creation Date</b>	18-02-2025
<b>Model Curriculum Valid Up to Date</b>	18-02-2028
<b>Model Curriculum Version</b>	4.0
<b>Minimum Duration of the Course</b>	420 Hours
<b>Maximum Duration of the Course</b>	420 Hours

## Program Overview

This section summarises the end objectives of the program along with its duration.

### Training Outcomes

At the end of the program, the learner should have acquired the listed knowledge and skills to:

- Explain the fundamental principles of designing algorithms and applying Object-Oriented Programming (OOP) concepts for scalable and maintainable code.
- Elucidate advanced programming techniques, including error handling, asynchronous programming, and API integration, to enhance software performance and stability.
- Describe the role of version control systems like Git and Agile practices such as sprint planning and retrospectives in collaborative software development.
- Discuss the importance of integrating Generative AI tools like GitHub Copilot for code generation, debugging, and task automation in software development.
- Enlist the best practices for ensuring software quality through static code analysis, debugging techniques, and performance optimization strategies.
- Discuss the Employability and Entrepreneurship Skills.

### Compulsory Modules

The table lists the modules and their duration corresponding to the Compulsory NOS of the QP.

NOS and Module Details	Theory Duration (Hours)	Practical Duration (Hours)	On-the-Job Training Duration (Mandatory) (Hours)	On-the-Job Training Duration (Recommended) (Hours)	Total Duration (Hours)
<b>SSC/N0506: Assist with Basic Software Development and Testing Tasks in the IT Services Industry</b> <b>NOS Version No.: 4.0</b> <b>NSQF Level: 4.0</b>	90:00	240:00	60:00	00:00	390:00
Module 1: Introduction to the IT-ITes Industry and the job role of a Junior Software Developer	05:00	00:00	00:00	00:00	05:00
Module 2: Software Basics and Development Lifecycle	10:00	25:00	10:00	00:00	45:00
Module 3: Problem-Solving with Advanced Coding Techniques	15:00	45:00	10:00	00:00	70:00
Module 4: Database Management and Version Control	15:00	45:00	10:00	00:00	70:00
Module 5: Agile Development and Collaboration	15:00	45:00	10:00	00:00	70:00

Module 6: Ensuring Code Quality and Deployment	15:00	45:00	10:00	00:00	70:00
Module 7: Security Practices and Generative AI Integration	15:00	35:00	10:00	00:00	60:00
<b>DGT/VSQ/N0101: Employability Skills (30 Hours) NOS Version No.: 1.0 NSQF Level: 2.0</b>	<b>30:00</b>	<b>00:00</b>	<b>00:00</b>	<b>00:00</b>	<b>30:00</b>
Module 8: Employability Skills (30 Hours)	30:00	00:00	00:00	00:00	30:00
<b>Total Duration</b>	<b>120:00</b>	<b>240:00</b>	<b>60:00</b>	<b>00:00</b>	<b>420:00</b>

# Module Details

## Module 1: Introduction to IT-ITeS Industry and Job Role of a Junior Software Developer

*Mapped to SSC/N0506, v4.0*

### Terminal Outcomes:

- Explain the importance of IT-ITeS Industry.
- Discuss the roles and responsibilities of a Junior Software Developer.

Duration (in hours): 05:00	Duration (in hours): 00:00
Theory – Key Learning Outcomes	Practical – Key Learning Outcomes
<ul style="list-style-type: none"> <li>• Define the IT-ITeS industry.</li> <li>• Describe the various sub-sectors within the IT-ITeS industry.</li> <li>• Discuss the scope of employment in the IT-ITeS industry.</li> <li>• Describe the roles and responsibilities of a Junior Software Developer for the software development activities by writing, testing, and debugging code using basic programming languages such as Python, Java, or JavaScript.</li> <li>• Discuss the basics of Agile development processes and Gen AI-based tools.</li> <li>• Discuss the future trends and career growth opportunities for a Junior Software Developer.</li> </ul>	-
Classroom Aids	
Training Kit - Facilitator’s Guide, Participant’s Handbook, Presentations and Software, Whiteboard, Marker, Projector, Laptop, Video Films	
Tools, Equipment and Other Requirements	
Nil	

## Module 2: Software Basics and Development Lifecycle

*Mapped to SSC/N0506, v4.0*

### Terminal Outcomes:

- Explain the fundamental principles of algorithm design and programming constructs.
- Discuss the Software Development Lifecycle (SDLC) and its various stages.

<b>Duration (in hours): 10:00</b>	<b>Duration (in hours): 25:00</b>
<b>Theory – Key Learning Outcomes</b>	<b>Practical – Key Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Explain the fundamental principles of algorithm design and programming constructs.</li> <li>• Describe the Software Development Lifecycle (SDLC) stages, including requirements gathering, design, development, testing, deployment, and maintenance.</li> <li>• Elucidate Object-Oriented Programming (OOP) concepts such as inheritance, encapsulation, and polymorphism, with their relevance to scalable and maintainable code.</li> <li>• Discuss best practices for writing clean, efficient, and optimized code in software development.</li> <li>• Explain the role of front-end technologies (HTML, CSS, JavaScript) in creating user-friendly interfaces.</li> <li>• Describe UI/UX principles and their application in designing intuitive and visually appealing applications.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate how to design and implement algorithms using appropriate programming constructs to solve simple problems.</li> <li>• Show how to apply Object-Oriented Programming (OOP) principles like inheritance and encapsulation to create reusable and maintainable code.</li> <li>• Demonstrate the ability to map real-world software requirements to different SDLC stages.</li> <li>• Show how to utilize front-end technologies (HTML, CSS, JavaScript) to build intuitive and responsive web interfaces.</li> <li>• Demonstrate the application of UI/UX principles by designing a simple, user-friendly interface prototype.</li> <li>• Show how to optimize code for performance and adherence to development best practices.</li> </ul>
<b>Classroom Aids</b>	
Training Kit - Facilitator’s Guide, Participant’s Handbook, Presentations and Software, Whiteboard, Marker, Projector, Laptop, Video Films	
<b>Tools, Equipment and Other Requirements</b>	
Code editor for HTML, CSS, JavaScript, and back-end programming (e.g. VSCodium, Brackets, etc.), Version Control System (e.g., GitLab, Gitea, etc.)	

## Module 3: Problem-Solving with Advanced Coding Techniques

*Mapped to SSC/N0506, v4.0*

### Terminal Outcomes:

- Describe debugging techniques to identify and resolve code issues.
- Explain how asynchronous programming can be implemented to optimize performance.

Duration (in hours): 15:00	Duration (in hours): 45:00
<b>Theory – Key Learning Outcomes</b>	<b>Practical – Key Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Explain the importance of debugging techniques and tools for resolving code issues.</li> <li>• Describe the role of asynchronous programming in enhancing application performance and responsiveness.</li> <li>• Elucidate the concept of error handling and its significance in ensuring software stability.</li> <li>• Discuss the use of APIs and web services for integrating external functionalities into applications.</li> <li>• Enlist common performance bottlenecks in software applications and techniques to address them.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate how to use debugging tools and techniques to identify and fix issues in a given codebase.</li> <li>• Show how to apply asynchronous programming methods, such as promises and async/await, for optimizing performance.</li> <li>• Demonstrate the integration of external functionalities into an application using APIs and web services.</li> <li>• Show how to implement robust error-handling mechanisms to enhance software stability and reliability.</li> <li>• Demonstrate techniques to identify performance bottlenecks and apply suitable optimizations to resolve them.</li> </ul>
<b>Classroom Aids</b>	
Training Kit - Facilitator’s Guide, Participant’s Handbook, Presentations and Software, Whiteboard, Marker, Projector, Laptop, Video Films	
<b>Tools, Equipment and Other Requirements</b>	
Version Control System (e.g., GitLab, Gitea, etc.)	

## Module 4: Database Management and Version Control

*Mapped to SSC/N0506, v4.0*

### Terminal Outcomes:

- Discuss how to develop and optimize SQL queries for efficient data handling.
- Elucidate the role of version control systems like Git/GitHub in managing code changes.

Duration (in hours): 15:00	Duration (in hours): 45:00
Theory – Key Learning Outcomes	Practical – Key Learning Outcomes
<ul style="list-style-type: none"> <li>• Explain the principles of database design, including schema creation and optimization.</li> <li>• Describe the role of SQL queries in managing and retrieving data efficiently.</li> <li>• Elucidate the use of ORM frameworks, such as Hibernate, for database operations and their benefits over traditional approaches.</li> <li>• Discuss the functionality and importance of version control systems, such as Git, in software development.</li> <li>• Enlist best practices for managing branches, commit messages, and collaborative workflows in Git/GitHub.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate how to write and optimize SQL queries for efficient data handling and retrieval.</li> <li>• Show how to use ORM frameworks like Hibernate to create, manage, and update database schemas.</li> <li>• Demonstrate the process of initializing and using Git for version control, including committing changes and creating repositories.</li> <li>• Show how to manage branches, resolve merge conflicts, and collaborate effectively on Git/GitHub projects.</li> <li>• Demonstrate the procedure for conducting code reviews and applying suggested improvements to enhance code quality.</li> </ul>
Classroom Aids	
Training Kit - Facilitator's Guide, Participant's Handbook, Presentations and Software, Whiteboard, Marker, Projector, Laptop, Video Films	
Tools, Equipment and Other Requirements	
Database Management System (e.g. MySQL, PostgreSQL, etc.), Version Control System (e.g., GitLab, Gitea, etc.)	

## Module 5: Agile Development and Collaboration

### Mapped to SSC/N0506, v4.0

#### Terminal Outcomes:

- Explain the key Agile ceremonies and their significance in project development.
- Describe how to break down user stories into actionable tasks for sprint execution.

Duration (in hours): 15:00	Duration (in hours): 45:00
Theory – Key Learning Outcomes	Practical – Key Learning Outcomes
<ul style="list-style-type: none"> <li>• Explain the core principles of Agile methodology and its benefits in software development.</li> <li>• Describe the purpose and format of Agile ceremonies, including sprint planning, daily standups, and retrospectives.</li> <li>• Elucidate the process of breaking down user stories into actionable tasks to streamline sprint execution.</li> <li>• Discuss the role of test case creation and execution in ensuring software functionality meets requirements.</li> <li>• Enlist effective techniques for documenting test results and collaborating with QA teams to address discrepancies and enhance coverage.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate participation in Agile ceremonies, such as sprint planning, standups, and retrospectives, to maintain team alignment and progress.</li> <li>• Show how to analyze user stories and break them down into smaller, actionable tasks for sprint execution.</li> <li>• Demonstrate the creation and execution of test cases to validate software functionality against defined requirements.</li> <li>• Show how to document test outcomes and communicate discrepancies effectively to the development team.</li> <li>• Demonstrate collaboration with QA teams to identify areas for test coverage improvement and ensure comprehensive validation.</li> </ul>
Classroom Aids	
Training Kit - Facilitator’s Guide, Participant’s Handbook, Presentations and Software, Whiteboard, Marker, Projector, Laptop, Video Films	
Tools, Equipment and Other Requirements	
Agile Project Management Tool (e.g. Taiga), Version Control System (e.g., GitLab, Gitea, etc.)	

## Module 6: Ensuring Code Quality and Deployment

*Mapped to SSC/N0506, v4.0*

### Terminal Outcomes:

- Elucidate the use of static analysis tools to ensure code quality.
- Describe deployment strategies like blue-green and canary deployments for efficient release management.

Duration (in hours): 15:00	Duration (in hours): 45:00
Theory – Key Learning Outcomes	Practical – Key Learning Outcomes
<ul style="list-style-type: none"> <li>• Explain the purpose and benefits of static analysis tools in maintaining code quality.</li> <li>• Describe the concepts and techniques of integration testing to ensure smooth component interaction.</li> <li>• Discuss various deployment strategies, including blue-green, canary, and rolling deployments, and their applications.</li> <li>• Elucidate the role of CI/CD pipelines in automating the software delivery process.</li> <li>• Enlist best practices for ensuring code quality, such as linting, refactoring, and code reviews.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate the use of static analysis tools and linting software to identify and resolve code quality issues.</li> <li>• Show how to perform integration testing using automated tools to verify seamless interactions between components.</li> <li>• Demonstrate the implementation of deployment strategies such as blue-green, canary, and rolling deployments in a controlled environment.</li> <li>• Show how to set up and manage a CI/CD pipeline to automate code integration, testing, and deployment processes.</li> <li>• Demonstrate the debugging of code issues and application of refactoring techniques to enhance maintainability.</li> </ul>
Classroom Aids	
Training Kit - Facilitator’s Guide, Participant’s Handbook, Presentations and Software, Whiteboard, Marker, Projector, Laptop, Video Films	
Tools, Equipment and Other Requirements	
Continuous Integration Tool (e.g. Jenkins, GitLab CI/CD, Concourse CI, etc.)	

## Module 7: Security Practices and Generative AI Integration

*Mapped to SSC/N0506, v4.0*

### Terminal Outcomes:

- Explain the importance of security measures like data encryption and vulnerability mitigation.
- Discuss the ethical considerations of using Generative AI in software development.

<b>Duration (in hours): 15:00</b>	<b>Duration (in hours): 35:00</b>
<b>Theory – Key Learning Outcomes</b>	<b>Practical – Key Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Explain the importance of data encryption and authentication mechanisms in securing software systems.</li> <li>• Describe common vulnerabilities like SQL injection and cross-site scripting (XSS), and strategies to mitigate them.</li> <li>• Discuss the ethical considerations of using Generative AI in software development, including biases and misuse risks.</li> <li>• Elucidate the capabilities, limitations, and potential of Generative AI tools in modern software development workflows.</li> <li>• Enlist best practices for prompt engineering to maximize the effectiveness of AI tools such as GitHub Copilot and ChatGPT.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate the implementation of fundamental security measures, including data encryption and authentication protocols.</li> <li>• Show how to identify and mitigate common vulnerabilities like SQL injection and cross-site scripting (XSS) using best practices.</li> <li>• Demonstrate the use of Generative AI tools like GitHub Copilot for coding, debugging, and optimizing code.</li> <li>• Show how to create precise and effective prompts to achieve desired outputs from Generative AI tools.</li> <li>• Demonstrate ethical use of Generative AI by aligning its application with development guidelines and constraints.</li> </ul>
<b>Classroom Aids</b>	
Training Kit - Facilitator’s Guide, Participant’s Handbook, Presentations and Software, Whiteboard, Marker, Projector, Laptop, Video Films	
<b>Tools, Equipment and Other Requirements</b>	
Continuous Integration Tool (e.g. Jenkins, GitLab CI/CD, Concourse CI, etc.)	

## Module 8: Employability Skills (30 Hours)

*Mapped to DGT/VSQ/N0101, v1.0*

**Duration: 30:00 Hours**

### Key Learning Outcomes

#### Introduction to Employability Skills Duration: 1 Hour

After completing this programme, participants will be able to:

1. Discuss the importance of Employability Skills in meeting the job requirements

#### Constitutional values - Citizenship Duration: 1 Hour

2. Explain constitutional values, civic rights, duties, citizenship, responsibility towards society etc. that are required to be followed to become a responsible citizen.

3. Show how to practice different environmentally sustainable practices

#### Becoming a Professional in the 21st Century Duration: 1 Hour

4. Discuss 21st-century skills.

5. Display a positive attitude, self-motivation, problem-solving, time management skills and continuous learning mindset in different situations.

#### Basic English Skills Duration: 2 Hours

6. Use appropriate basic English sentences/phrases while speaking

#### Communication Skills Duration: 4 Hours

7. Demonstrate how to communicate in a well-mannered way with others.

8. Demonstrate working with others in a team

#### Diversity & Inclusion Duration: 1 Hour

9. Show how to conduct oneself appropriately with all genders and PwD

10. Discuss the significance of reporting sexual harassment issues in time

#### Financial and Legal Literacy Duration: 4 Hours

11. Discuss the significance of using financial products and services safely and securely.

12. Explain the importance of managing expenses, income, and savings.

13. Explain the significance of approaching the concerned authorities in time for any exploitation as per legal rights and laws

#### Essential Digital Skills Duration: 3 Hours

14. Show how to operate digital devices and use the associated applications and features, safely and securely

15. Discuss the significance of using the internet for browsing, and accessing social media platforms, safely and securely



**Entrepreneurship Duration: 7 Hours**

16. Discuss the need for identifying opportunities for potential business, sources for arranging money and potential legal and financial challenges

**Customer Service Duration: 4 Hours**

17. Differentiate between types of customers

18. Explain the significance of identifying customer needs and addressing them

19. Discuss the significance of maintaining hygiene and dressing appropriately

**Getting ready for Apprenticeship & Jobs Duration: 2 Hours**

20. Create a biodata

21. Use various sources to search and apply for jobs

22. Discuss the significance of dressing up neatly and maintaining hygiene for an interview

23. Discuss how to search and register for apprenticeship opportunities

## Module 9: On-the-Job Training

### Mapped to Junior Software Developer

<b>Mandatory Duration (in hours): 60:00</b>	<b>Recommended Duration (in hours): 00:00</b>
<b>Location: On-Site</b>	
<p><b>Terminal Outcomes</b></p> <ul style="list-style-type: none"> <li>• Demonstrate the design and implementation of algorithms using programming constructs.</li> <li>• Show how to apply Object-Oriented Programming principles to create scalable code.</li> <li>• Demonstrate debugging techniques to identify and fix code issues.</li> <li>• Show how to optimize code for performance and adhere to best practices.</li> <li>• Demonstrate the use of APIs and web services for integrating external functionalities.</li> <li>• Show how to develop efficient SQL queries and manage database schemas.</li> <li>• Demonstrate using Git or GitHub for version control and managing code changes.</li> <li>• Show how to participate in Agile ceremonies and collaborate on user stories.</li> <li>• Demonstrate drafting and executing test cases to verify software functionality.</li> <li>• Show how to document test outcomes and report discrepancies effectively.</li> <li>• Demonstrate performing code reviews and refactoring code for maintainability.</li> <li>• Show how to use automated integration testing tools to verify system interactions.</li> <li>• Demonstrate executing deployment strategies like rolling and blue-green deployments.</li> <li>• Show how to apply security measures to prevent common vulnerabilities.</li> <li>• Demonstrate the basics of UI/UX design and create responsive interfaces.</li> <li>• Show how to use CI/CD pipelines for automating testing and deployment.</li> <li>• Demonstrate utilizing Generative AI tools for coding and debugging tasks.</li> <li>• Show how to create effective prompts to enhance AI-generated outputs.</li> <li>• Demonstrate understanding and implementing the stages of the software development lifecycle.</li> <li>• Show how to collaborate in SDLC models such as Agile or DevOps.</li> </ul>	

## Annexure

### Trainer Requirements

1.	<b>Trainer's Qualification and experience in the relevant sector (in years) (as per NCVET guidelines)</b>	<p><b>Educational Qualification:</b> Graduate in any discipline</p> <p><b>Industry &amp; Training Experience:</b> 2 years of industry experience in software development roles.</p> <p><b>Certification:</b> "Trainer" mapped to the Qualification Pack "MEP/Q2601" Minimum accepted score is 80% aggregate.</p>
2.	<b>Master Trainer's Qualification and experience in the relevant sector (in years) (as per NCVET guidelines)</b>	<p><b>Educational Qualification:</b> Graduate in any discipline</p> <p><b>Industry &amp; Training Experience:</b> 4 years of industry experience in software development roles.</p> <p><b>Certification:</b> "Master Trainer" mapped to the Qualification Pack "MEP/Q2602" Minimum accepted score is 90% aggregate.</p>
3.	<b>Tools and Equipment Required for Training</b>	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No (If "Yes", details to be provided in Annexure)
4.	<b>In Case of Revised Qualification, Details of Any Upskilling Required for Trainer</b>	NA

## Assessor Requirements

1.	<b>Assessor's Qualification and experience in relevant sector (in years) (as per NCVET guidelines)</b>	<p><b>Educational Qualification:</b> Graduate in any discipline</p> <p><b>Industry &amp; Training Experience:</b> 2 years of industry experience in software development roles.</p> <p><b>Certification:</b> "Assessor" mapped to the Qualification Pack "MEP/Q2701" Minimum accepted score is 80% aggregate.</p>
2.	<b>Proctor's Qualification and experience in relevant sector (in years) (as per NCVET guidelines), (wherever applicable)</b>	<p><b>Educational Qualification:</b> Graduate in any discipline</p> <p><b>Industry &amp; Training Experience:</b> 2 years of industry experience in software development roles.</p> <p><b>Certification:</b> "Assessor" mapped to the Qualification Pack "MEP/Q2701" Minimum accepted score is 80% aggregate.</p>
3.	<b>Lead Assessor's/Proctor's Qualification and experience in relevant sector (in years) (as per NCVET guidelines)</b>	<p><b>Educational Qualification:</b> Graduate in any discipline</p> <p><b>Industry &amp; Training Experience:</b> 4 years of industry experience in software development roles.</p> <p><b>Certification:</b> "Lead Assessor" mapped to the Qualification Pack "MEP/Q2702" Minimum accepted score is 90% aggregate.</p>
4.	<b>Assessment Mode (Specify the assessment mode)</b>	The assessment will consist of a blend of hands-on practical evaluations, viva-voce, and online proctored scenario-based multiple-choice questions ensuring a thorough evaluation of the individual's proficiency in learning outcomes, practical understanding, and real-world application of concepts.
5.	<b>Tools and Equipment Required for Assessment</b>	<input checked="" type="checkbox"/> Same as for training <input type="checkbox"/> Yes <input type="checkbox"/> No (details to be provided in Annexure-if it is different for Assessment)

## Assessment Strategy

This section includes the processes involved in identifying, gathering and interpreting information to evaluate the learner on the required competencies of the program.

### Batch Creation & Assessment Request:

Training Providers (TP) or Training Centers (TC), including any other authorized partner of Ministry/ Department create batches / push batches on the SIDH portal. Assessment requests are submitted through the SIDH portal or via email or other media as authorized from time to time. For NON-SIDH schemes, assessment requests are received electronically or through respective State Skill Mission portals. TP/TC initiates the assessment request through the InSDMS portal and processes the payment (where applicable).

### Batch Alignment & Confirmation:

Upon payment confirmation, batches are assigned to the Assessment Agency based on factors like:

- Assessment readiness
- Availability of certified assessors for the specific job role
- Assessment capping to an assessment agency as prescribed from time to time for an AB An email communication / prescribed mode communication is sent to TP/TC for confirmation of the assessment date, with IT-ITeS SSC in the loop. Once confirmation is received, the Assessment Agency designates a TOA-certified assessor to conduct or facilitate the assessment.
- Batches are only formed when the Qualification is active.

### Candidate Verification & Assessment Execution:

Candidate details are verified and documented at the beginning of the assessment by a certified assessor. A Quality Assurance (QA) mechanism is enforced, requiring an undertaking from the TC. Regular feedback is collected from TP/TC to ensure continuous improvement.

### Evidence Collection & Validation:

Proctors or assessors capture date/time-stamped and geo-tagged photographs of the assessment location during the process. Attendance is also ensured offline. A PC-wise result analysis is conducted to refine assessment standards.

### Monitoring & Compliance:

Batch monitoring follows established protocols, ensuring adherence to assessment guidelines. Sample based surprise visits are conducted at TC locations during both training and assessments to verify compliance. This structured approach ensures transparency, quality control, and validation throughout the assessment process.

### Testing Environment:

- Check the Assessment location, date and time
- If the batch size is more than 30, then there should be 2 Assessors.
- Check that the allotted time to the candidates to complete Theory & Practical Assessment is correct.

### Assessment Quality Assurance levels/Framework:

IT-ITeS SSC nasscom is responsible for the development and periodic review of the question bank developed for a specific job role. We publish an openly accessible sample /model question paper on our website for all stakeholders. The quality of the Question Bank created by the assessment designer is validated by a Subject matter experts on the following parameters:

- Appropriateness of the Question Bank in terms of facts, data and information.
- Checks for grammar, spellings, scripting and formatting.
- The information provided should be specific enough to remove any ambiguity in answers/solutions to the question.
- Relevance – Assessing the topic well w.r.t. the job role.
- Check if the difficulty level of each question is as per the matrix.
- Check if the images used in the question are clear and relevant.
- All variables, symbols and abbreviations used must be declared.
- The correct answer option should be unique, and the options should not be overlapping

## References

### Glossary

Term	Description
<b>Declarative Knowledge</b>	Declarative knowledge refers to facts, concepts and principles that need to be known and/or understood in order to accomplish a task or to solve a problem.
<b>Key Learning Outcome</b>	Key learning outcome is the statement of what a learner needs to know, understand and be able to do in order to achieve the terminal outcomes. A set of key learning outcomes will make up the training outcomes. Training outcome is specified in terms of knowledge, understanding (theory) and skills (practical application).
<b>OJT (M)</b>	On-the-job training (Mandatory); trainees are mandated to complete specified hours of training on site
<b>OJT (R)</b>	On-the-job training (Recommended); trainees are recommended the specified hours of training on site
<b>Procedural Knowledge</b>	Procedural knowledge addresses how to do something, or how to perform a task. It is the ability to work, or produce a tangible work output by applying cognitive, affective, or psychomotor skills.
<b>Training Outcome</b>	Training outcome is a statement of what a learner will know, understand and be able to do it upon the completion of the training.
<b>Terminal Outcome</b>	Terminal outcome is a statement of what a learner will know, understand and be able to do upon the completion of a module. A set of terminal outcomes help to achieve the training outcome.

## Acronyms and Abbreviations

Term	Description
NCVET	National Council for Vocational Education and Training
QP	Qualification Pack
MC	Model Curriculum
NSQF	National Skills Qualification Framework
NSQC	National Skills Qualification Committee
NOS	National Occupational Standards
NCO	National Classification of Occupations
ES	Employability Skills
OJT	On the Job Training
NASSCOM	National Association of Software and Service Companies
IT-ITeS	Information Technology and Information Technology Enabled Services
SDLC	SDLC – Software Development Lifecycle
OOP	OOP – Object-Oriented Programming
API	API – Application Programming Interface
CI/CD	CI/CD – Continuous Integration/Continuous Deployment
AI	AI – Artificial Intelligence
UI/UX	UI/UX – User Interface/User Experience
ORM	ORM – Object-Relational Mapping
SQL	SQL – Structured Query Language
DevOps	DevOps – Development and Operations
XSS	XSS – Cross-Site Scripting
Git	Git – Global Information Tracker
HTML	HTML – HyperText Markup Language
CSS	CSS – Cascading Style Sheets
Gen AI	Gen AI – Generative Artificial Intelligence
IDE	IDE – Integrated Development Environment
DBMS	DBMS – Database Management System
VCS	VCS – Version Control System