CONTENTS

Chapter 1 Internship Organisation Profile

Chapter 2 Executive Summary

Chapter 3 Internship Part

Chapter 4 Outcomes Description

Chapter 5 Photos and Video Links

Chapter 6 Activity Log

- Week 1 — Introduction to Quantum Computing Approaches
- Week 2 — Introduction to Qiskit and Single-Qubit Gates
- Week 3 — Rotation Gates, Multi-Qubit Systems and Tensor Products
- Week 4 — Entanglement, Bell States and Advanced Gates
- Week 5 — Introduction to Quantum Algorithms
- Week 6 — Deutsch, Deutsch-Jozsa and Bernstein-Vazirani Algorithms
- Week 7 — Simon's Algorithm and Grover's Algorithm
- Week 8 — Final Project, Documentation and Submission

CHAPTER 1

Internship Organisation Profile

1. Introduction

Anjanaas Foundation is a registered non-profit organisation based in Avanigadda, Krishna District, Andhra Pradesh. The foundation is dedicated to empowering individuals through education, skill development, health, agriculture, and community development. Built upon a decade-long legacy of educational service, Anjanaas Foundation has expanded its mission to address social protection and financial independence for the most vulnerable sections of society. The foundation plays a significant role in fostering experiential learning through its well-structured internship programmes.

2. Vision

To create a self-sustaining society by providing quality education, skill development, and opportunities for holistic growth, ensuring that every individual can contribute meaningfully to the community.

3. Mission

- To bridge the gap between education and employment by equipping individuals with industry-relevant skills.
- To provide inclusive learning environments that cater to diverse communities, including underprivileged and differently-abled individuals.
- To promote innovation, entrepreneurship, and sustainability through practical training programmes.

4. Objectives

- To offer structured internship programmes that provide hands-on experience in various disciplines.
- To collaborate with educational institutions to integrate internships into academic curricula.
- To enhance employability by offering training in high-demand skills such as IT, business management, agriculture, and life sciences.
- To promote research and development through skill-based learning initiatives.

5. Internship Programme Overview

a. Internship Scope and Reach

Anjanaas Foundation has provided more than 10,000 internship opportunities across various universities in Andhra Pradesh. The foundation collaborates with institutions such as Krishna University to formalise internship programmes through Memoranda of Understanding (MoUs), ensuring academic recognition for participating students.

b. Key Training Areas

Information Technology and Digital Skills

Python, Data Science, Cloud Computing, Full Stack Development, Digital Marketing, DevOps, ServiceNow, MySQL, Tableau, Web Technologies, Generative AI, Machine Learning, Agentic AI.

Business and Management

Office Administration, Tally, SAP, Competitive Exam Preparation, Tourism.

Science and Agriculture

Life Sciences, Agriculture, Forestry, Pharmacy, Food Technology.

c. Industry Collaboration and Career Development

Anjanaas Foundation works with leading companies across India, ensuring that trained interns receive priority consideration in recruitment drives. Practical learning experiences prepare interns for real-world challenges, enhancing their employability and professional readiness.

CHAPTER 2

Executive Summary

This internship programme focused on the study and practical application of Quantum Computing concepts and technologies using the Qiskit framework developed by IBM. The training was conducted under the guidance of Anjanaas Foundation, where students were introduced to both the theoretical foundations of quantum computing and hands-on circuit implementation using Qiskit.

The programme commenced with a conceptual introduction to quantum computing, covering the fundamental differences between classical and quantum computation, the physical principles underlying qubit behaviour, and an overview of major quantum computing approaches including superconducting qubits, trapped ions, and photonic systems. Students were introduced to the Qiskit framework and learned how to construct and simulate quantum circuits using the AerSimulator backend.

The core training sessions progressed through single-qubit gates including the Pauli X, Y, and Z gates and the Hadamard gate, followed by rotation gates RX, RY, and RZ that enable continuous angular control over qubit states on the Bloch sphere. Students then studied multi-qubit systems and the mathematical framework of tensor products that describes how individual qubit states combine to form composite quantum systems.

The programme extended to quantum entanglement and the four Bell states, demonstrating how entangled qubit pairs exhibit non-classical correlations. Advanced gate operations including the S gate, S-dagger gate, T gate, T-dagger gate, and three-qubit GHZ state preparation were also covered, providing students with a comprehensive understanding of the standard quantum gate library.

The latter phase of the programme focused on quantum algorithms, beginning with an introduction to the principles that give quantum algorithms their computational advantage, followed by detailed study of the Deutsch Algorithm, Deutsch-Jozsa Algorithm, Bernstein-Vazirani Algorithm, Simon's Algorithm, and Grover's Search Algorithm. Each algorithm was studied both theoretically and through Qiskit circuit implementation.

The programme culminated in the development of a final mini project titled "Quantum Random Number Generator (QRNG) Using Qiskit," in which students designed and implemented a complete QRNG pipeline that exploits quantum superposition to generate genuinely random bits, formats them into usable number types, validates their statistical properties, and demonstrates practical applications.

Sector of Development

This internship falls under the Quantum Computing and Emerging Technologies sector, with a focus on quantum circuit design, quantum gate operations, quantum algorithm implementation, and quantum software development. The training covered multiple technological domains including quantum mechanics fundamentals, Qiskit circuit programming, quantum algorithm theory, and quantum application development.

Learning Objectives

- To understand the fundamental principles of quantum computing and its distinction from classical computing.
- To learn the quantum computing landscape including different hardware approaches and their characteristics.
- To gain proficiency in constructing and simulating quantum circuits using the Qiskit framework.
- To understand single-qubit gate operations including Pauli gates, Hadamard, and rotation gates.
- To study multi-qubit systems, tensor products, and the scaling properties of quantum state spaces.

• To understand quantum entanglement and the construction of Bell states and GHZ states.

• To learn the theoretical foundations of quantum algorithm design and quantum speedup.

• To implement and simulate landmark quantum algorithms including Deutsch, Deutsch-Jozsa, Bernstein-Vazirani, Simon, and Grover algorithms.

• To apply quantum programming knowledge in the development of a complete mini project.

Outcomes Achieved

• Developed a clear understanding of quantum computing principles and the behaviour of qubits.

• Gained practical experience in constructing and executing quantum circuits using Qiskit and AerSimulator.

• Learned to apply single-qubit and multi-qubit gate operations in circuit design.

• Understood quantum entanglement and implemented all four Bell states and the GHZ state.

• Acquired knowledge of quantum algorithm design and the principle of quantum parallelism.

• Successfully implemented Deutsch, Deutsch-Jozsa, Bernstein-Vazirani, Simon, and Grover algorithms in Qiskit.

• Designed and developed a complete Quantum Random Number Generator as a final project.

• Improved analytical reasoning, logical thinking, and technical documentation skills.

CHAPTER 3

Internship Part

Description of Activities and Responsibilities

The internship focused on developing a thorough understanding of quantum computing theory and practical quantum circuit programming using Qiskit. Students progressed from foundational quantum mechanics concepts through to the implementation of advanced quantum algorithms and the development of a complete quantum application project.

The activities carried out during the internship included studying quantum computing approaches and hardware technologies, learning the Qiskit circuit model, implementing quantum gates and circuit experiments, studying entanglement and multi-qubit systems, implementing landmark quantum algorithms, and developing the final QRNG mini project.

Work Environment and Tools Used

The internship was conducted in an online learning environment organised by Anjanaas Foundation, where students participated in structured training sessions delivered through live classes and YouTube video lectures. Sessions were interactive and combined theoretical explanation with practical circuit demonstration.

Tool / Technology	Purpose
Programming Language	Python 3
Quantum Computing Framework	Qiskit (IBM Quantum)
Simulation Backend	Qiskit Aer — AerSimulator
Development Environment	Google Colab (Cloud-based Python and Qiskit environment)
Supporting Library	NumPy (for statevector analysis)
Visualisation	Qiskit circuit_drawer, measurement histograms
Session Platform	Zoom (live classes) and YouTube (recorded lectures)

Weekly Work Schedule Overview

The internship followed an eight-week structured training schedule designed to provide progressive learning from foundational concepts to advanced quantum algorithm implementation.

Weeks 1 – 2 — Foundations

Introduction to quantum computing approaches and hardware technologies, followed by Introduction to Qiskit and foundational single-qubit gate operations.

Weeks 3 – 6 — Core Training

Rotation gates and Bloch sphere representation; multi-qubit systems and tensor products; quantum entanglement, Bell states, and advanced gate operations; introduction to quantum algorithms and the quantum algorithm landscape.

Weeks 7 – 8 — Algorithms and Project

Implementation of Deutsch, Deutsch-Jozsa, Bernstein-Vazirani, Simon, and Grover algorithms; development of the Quantum Random Number Generator mini project; documentation, logbook completion, and submission.

Tasks Performed

• Studying quantum computing principles and comparing classical and quantum computational models.
• Learning about qubit hardware technologies including superconducting qubits, trapped ions, and photonic approaches.
• Constructing and executing quantum circuits using Qiskit's QuantumCircuit API.
• Implementing single-qubit gate operations including X, Y, Z, H, S, S-dagger, T, and T-dagger gates.
• Applying rotation gates RX, RY, and RZ to manipulate qubit states on the Bloch sphere.
• Building multi-qubit circuits and understanding tensor product composition of quantum states.
• Constructing all four Bell states and verifying entanglement through measurement correlation.
• Preparing and verifying three-qubit GHZ entangled states.
• Studying and implementing the Deutsch, Deutsch-Jozsa, and Bernstein-Vazirani algorithms.
• Implementing Simon's Algorithm and Grover's Search Algorithm with oracle circuits.
• Designing and developing the Quantum Random Number Generator mini project.
• Preparing project documentation, internship logbook, and final submission materials.

Skills Acquired

Quantum Computing Fundamentals

Understanding qubit states, superposition, measurement, and the distinction between quantum and classical information processing.

Qiskit Circuit Programming

Proficiency in constructing, simulating, and analysing quantum circuits using the Qiskit framework and AerSimulator backend.

Quantum Gate Operations

Ability to apply and interpret the action of standard single-qubit and multi-qubit quantum gate operations in circuit design.

Quantum Entanglement

Understanding of quantum entanglement theory and practical ability to construct entangled states including Bell states and GHZ states.

Quantum Algorithm Design

Knowledge of landmark quantum algorithms and the quantum circuit structures that realise computational speedups over classical methods.

Quantum Application Development

Experience in designing and implementing a complete quantum application from problem formulation through circuit design to result analysis.

Reflection

The internship provided a comprehensive and structured introduction to one of the most significant emerging technologies of the present era. The progression from quantum computing fundamentals through gate operations to algorithm implementation created a coherent learning pathway that built both conceptual understanding and practical skill.

The hands-on nature of the training, with every theoretical concept reinforced through Qiskit circuit implementation, enabled students to move beyond passive understanding to active engagement with quantum systems. Working with real quantum algorithms such as Grover's search and Simon's algorithm provided direct insight into how quantum mechanics can be harnessed for computational advantage.

The development of the QRNG final project brought together all the skills developed during the programme — circuit design, superposition, measurement, and statistical analysis — into a single coherent application. Overall, the internship significantly strengthened both technical competency in quantum computing and professional skills in systematic learning, documentation, and project execution.

CHAPTER 4

Outcomes Description

Work Environment Experience

The internship was conducted in an online learning environment organised through the training programmes of Anjanaas Foundation. The environment emphasised disciplined learning, systematic training, and collaborative knowledge sharing among participants.

Students interacted regularly with faculty members through guided live sessions conducted on Zoom and supplementary recorded lectures available on YouTube. This combination of live interaction and self-paced revision created a flexible yet structured learning atmosphere. Students could clarify doubts during live sessions while revisiting complex circuit demonstrations through the recorded content.

Although the internship was conducted remotely, the programme maintained a professional work culture that emphasised punctuality, consistent participation, and adherence to the weekly learning schedule. The structured progression from foundations through core training to project development encouraged students to develop discipline and responsibility in their learning approach.

The availability of topic-specific lectures for each class session — covering quantum gates, entanglement, and algorithms — ensured that students could engage with content at their own pace alongside the live instruction. Regular communication with faculty ensured that students received timely guidance throughout the programme.

Real-Time Technical Skills Acquired

The internship helped students acquire several technical skills in quantum computing and quantum software development that are directly relevant to industry and research in this rapidly advancing field.

Students gained a strong conceptual foundation in quantum computing, including an understanding of qubit state representation, quantum superposition, measurement collapse, and the Bloch sphere model. The distinction between quantum and classical information processing was made concrete through comparative analysis of specific computational problems.

Practical proficiency was developed in Qiskit, IBM's open-source quantum computing framework. Students learned to construct quantum circuits using the QuantumCircuit API, apply gate operations, transpile circuits for the AerSimulator backend, execute simulations, and retrieve and interpret measurement results.

A comprehensive understanding of the standard quantum gate library was developed, covering the Pauli X, Y, and Z gates, the Hadamard gate, the CNOT gate, rotation gates, phase gates including S and T, and their conjugate transposes. Students learned to predict the action of each gate on qubit states both mathematically and through circuit simulation.

The study of quantum algorithms provided students with insight into how quantum parallelism, phase kickback, amplitude amplification, and interference are harnessed to achieve computational speedups. Implementing these algorithms in Qiskit connected theoretical understanding with practical circuit construction.

Managerial Skills Acquired

In addition to technical knowledge, the internship helped students develop several managerial and professional competencies. Students learned to plan their weekly learning activities according to the structured programme schedule, developing time management and task prioritisation skills.

The development of the final QRNG project required goal-oriented planning, systematic problem decomposition, and disciplined execution across multiple stages including circuit design, coding, testing, and documentation. These activities developed students' ability to manage a technical project from conception to completion within a defined timeframe.

Students also developed analytical thinking and problem-solving abilities while designing oracle circuits for quantum algorithms and debugging circuit behaviour through measurement analysis. Evaluating the outputs of quantum simulations and comparing them to theoretical expectations strengthened their capacity for systematic reasoning.

Communication Skills Development

The internship contributed significantly to the improvement of communication skills in both written and verbal forms. Students participated actively in live session discussions, asking questions about circuit design, gate action, and algorithm logic. These interactions improved confidence in articulating complex technical concepts in accessible language.

Written communication skills were enhanced through the preparation of the internship logbook and the final project documentation. Students learned to present quantum computing concepts, circuit descriptions, algorithm explanations, and project results in a clear, structured, and professionally formatted manner.

Participation in Team Discussions and Collaborative Learning

Collaborative learning played an important role throughout the programme. Students engaged in group discussions during live sessions about quantum circuit design, algorithm construction, and the interpretation of measurement outcomes. These discussions enabled participants to explore multiple approaches to circuit problems and develop a deeper understanding of quantum system behaviour through shared reasoning.

Technological Developments Observed During the Internship

During the internship, students observed several important developments in the field of quantum computing that are shaping the direction of the industry and research.

One of the most significant developments is the rapid progress in quantum hardware, with IBM, Google, and other organisations achieving increasing qubit counts and improving qubit coherence times. The emergence of error-corrected logical qubits represents a major milestone on the path to fault-tolerant quantum computation.

Students also observed the growing importance of quantum software ecosystems, with frameworks such as Qiskit enabling developers to work with quantum computers without requiring deep expertise in experimental physics. This democratisation of quantum computing tools is accelerating research and application development globally.

The increasing focus on hybrid quantum-classical algorithms, where quantum subroutines are integrated into classical computational workflows, was also noted as a key direction for near-term quantum applications in optimisation, chemistry simulation, and machine learning.

Conclusion

The Quantum Technologies using Qiskit internship provided valuable exposure to both the theoretical foundations and practical implementation of quantum computing technologies. Throughout the programme, students progressed from understanding basic quantum mechanics and qubit behaviour to constructing complex quantum algorithm circuits and developing a complete quantum application.

The hands-on approach to learning, with every concept reinforced through Qiskit circuit implementation and AerSimulator execution, enabled students to develop genuine practical competency in quantum programming. The final QRNG project demonstrated how quantum superposition and measurement can be applied to solve a real-world problem with direct relevance to cryptography and secure system design.

Overall, the internship strengthened technical skills, analytical thinking, and systematic problem-solving while providing practical insight into one of the most important and rapidly evolving fields in modern computing. The knowledge and experience gained will serve as a strong foundation for further study or career development in quantum computing and related technologies.

CHAPTER 5

Photos and Video Links

Photographs

Search for images and screenshots related to the topics covered in this internship programme — quantum circuits, gate diagrams, Bloch sphere representations, algorithm flowcharts, Qiskit simulation outputs, and project screenshots. Print them in colour, cut, and paste them in the space below and on the following pages of this section.

Video Links

All class sessions for this internship programme were recorded and uploaded to the Anjanaas Foundation YouTube channel as a playlist. Students are encouraged to use these recordings for revision and reference.

YouTube Course Playlist :
https://youtube.com/playlist?list=PLx3CGw_4F7Gy_aGq53CiPvR3FtuysAhmw&si;=OPJcvKrfyO-WNtNj

Zoom Meeting Link

Live training sessions for this programme were conducted through Zoom. The meeting link is provided below.

Zoom Meeting Link :
https://us06web.zoom.us/j/87542788545?pwd=IlQva0UJO5CTyQyzzwW63sj6fcT7O4.1

CHAPTER 6

Activity Log

WEEK 1 — Introduction to Quantum Computing Approaches

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Overview of classical computing and its limitations for certain problem classes	Understanding why quantum computing is needed	Faculty
Day 2	Principles of quantum mechanics relevant to computing: superposition and measurement	Understanding qubit behaviour and probabilistic outcomes	Faculty
Day 3	Study of major quantum computing hardware approaches: superconducting qubits and trapped ions	Learning the physical basis of different qubit technologies	Faculty
Day 4	Study of photonic and neutral-atom quantum computing approaches	Understanding alternative hardware paradigms	Faculty
Day 5	Overview of the quantum computing industry landscape and leading organisations	Awareness of the global quantum ecosystem	Faculty
Day 6	Introduction to the quantum software stack and the role of Qiskit	Understanding how software connects to quantum hardware	Faculty

Objective: To introduce the fundamental principles of quantum computing, its distinction from classical computing, and the major hardware approaches used to realise quantum systems.

Detailed Report

The first week of the internship began with a broad introduction to quantum computing and the motivations behind its development. Students were introduced to the fundamental limitations of classical computing, particularly the challenge posed by problems whose solution space grows exponentially with input size. This discussion established the context for why quantum computing is being pursued as a transformative technology.

The core quantum mechanical principles underlying quantum computing were introduced, including the concept of the qubit as a two-level quantum system, the principle of superposition that allows a qubit to exist in a combination of basis states before measurement, and the probabilistic nature of quantum measurement outcomes governed by the Born rule.

Students then studied the major hardware approaches to realising quantum computers. Superconducting qubit systems, which operate at temperatures near absolute zero and are used by IBM and Google, were examined alongside trapped-ion systems, which use electrically confined atomic ions as qubits. Photonic and neutral-atom approaches were also discussed, providing students with a broad view of the quantum hardware landscape.

The week concluded with an overview of the quantum computing software stack, explaining how high-level programming frameworks such as Qiskit translate abstract circuit descriptions into instructions that can be executed on physical quantum hardware or simulated on classical computers.

WEEK 2 — Introduction to Qiskit and Single-Qubit Gates

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Introduction to Qiskit: QuantumCircuit API, quantum and classical registers	Understanding the basic structure of a Qiskit circuit	Faculty
Day 2	Pauli X gate: bit-flip operation, matrix representation, and circuit application	Learning the quantum NOT gate and its action on qubit states	Faculty
Day 3	Pauli Y and Z gates: phase and combined operations, Bloch sphere interpretation	Understanding phase and combined gate operations	Faculty
Day 4	Hadamard gate: superposition creation, equal probability states, circuit use	Understanding how the H gate creates quantum superposition	Faculty
Day 5	Measurement in Qiskit: measurement operators, result histograms, shot statistics	Learning how to read and interpret quantum measurement outputs	Faculty
Day 6	Practical: building and simulating circuits with all single-qubit gates using AerSimulator	Hands-on circuit construction and result verification	Faculty

Objective: To introduce the Qiskit framework and develop practical proficiency in constructing and simulating quantum circuits using the fundamental single-qubit gate operations.

Detailed Report

The second week introduced students to Qiskit, IBM's open-source quantum computing framework, beginning with the QuantumCircuit API. Students learned how to create quantum circuits by specifying the number of quantum and classical registers, applying gate operations to individual qubits, and adding measurement instructions that map qubit states to classical bit outcomes.

The Pauli X gate, the quantum analogue of the classical NOT gate, was the first gate studied. Students learned its matrix representation and observed its action on the computational basis states. The Pauli Y and Z gates were then introduced, with discussion of their phase-rotation properties and geometric interpretation on the Bloch sphere.

The Hadamard gate was introduced as one of the most important gates in quantum computing, as it transforms a qubit initialised in the ground state into an equal superposition of the zero and one basis states. Students verified this behaviour by executing Hadamard circuits on the AerSimulator and observing the resulting measurement histograms, which showed approximately equal counts for both outcomes.

The week concluded with a practical session in which students built circuits incorporating all four single-qubit gates covered, executed them on the AerSimulator, and compared the observed measurement distributions with theoretical predictions. This hands-on session established the foundational circuit-building workflow that students would use throughout the remainder of the programme.

WEEK 3 — Rotation Gates, Multi-Qubit Systems and Tensor Products

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	RX gate: rotation about X-axis of Bloch sphere, parameterised angle, matrix form	Understanding continuous rotation of qubit state	Faculty
Day 2	RY and RZ gates: rotations about Y and Z axes, phase accumulation	Learning the full set of single-axis rotation operations	Faculty
Day 3	Introduction to multi-qubit systems: two-qubit state space and basis states	Understanding how qubit state spaces combine	Faculty
Day 4	Tensor product mathematics: combining qubit state vectors	Learning the mathematical framework for composite quantum states	Faculty
Day 5	Quantum scaling: exponential growth of state space with number of qubits	Understanding why quantum systems scale beyond classical simulation	Faculty
Day 6	Practical: implementing rotation gate circuits and two-qubit state analysis in Qiskit	Applying rotation gates and multi-qubit concepts in circuit design	Faculty

Objective: To understand continuous qubit rotation operations through the RX, RY, and RZ gates, and to develop a mathematical and conceptual understanding of multi-qubit quantum systems using tensor products.

Detailed Report

The third week opened with a study of the rotation gates, which extend the discrete gate operations covered in the previous week to continuous angular rotations. The RX gate performs a rotation of the qubit state about the X-axis of the Bloch sphere by a specified angle. Students studied its parameterised matrix form and observed how varying the rotation angle produces a continuum of qubit states between the computational basis states.

The RY and RZ gates were then introduced, covering rotations about the Y-axis and phase accumulation about the Z-axis respectively. Students learned that any single-qubit unitary operation can be decomposed into a sequence of these three rotation gates, making them the building blocks of universal single-qubit control.

The second half of the week introduced multi-qubit systems. Students learned that the state space of a system of n qubits is described by a vector in a 2^n-dimensional Hilbert space, constructed as the tensor product of the individual qubit state spaces. The four computational basis states of a two-qubit system were derived and their physical interpretation discussed.

The tensor product formalism was studied as the mathematical mechanism by which individual qubit states combine into joint system states. Students calculated tensor products of example qubit states and verified that product states could be factored back into their component parts. The exponential scaling of the quantum state space with qubit number was highlighted as the origin of the computational advantage that quantum systems possess over classical simulators.

WEEK 4 — Entanglement, Bell States and Advanced Gate Operations

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	CNOT gate: controlled-NOT operation, truth table, and circuit symbol	Understanding two-qubit conditional gate operations	Faculty
Day 2	Quantum entanglement: definition, EPR pairs, non-local correlations	Understanding the nature and significance of entanglement	Faculty
Day 3	Bell state construction: creating all four Bell states using H and CNOT	Learning to construct maximally entangled two-qubit states	Faculty
Day 4	S gate, S-dagger gate and T gate: phase gate operations and their inverses	Understanding quarter-turn and eighth-turn phase gates	Faculty
Day 5	T-dagger gate and three-qubit GHZ state preparation	Learning to prepare multi-qubit entangled states	Faculty
Day 6	Practical: implementing Bell states and GHZ state circuits, verifying measurement correlations	Hands-on verification of entanglement through correlated measurement outcomes	Faculty

Objective: To understand quantum entanglement, construct the four Bell states and the three-qubit GHZ state, and study the S, S-dagger, T, and T-dagger phase gate operations.

Detailed Report

The fourth week introduced quantum entanglement, one of the most distinctive and counterintuitive features of quantum mechanics. Students learned that entangled quantum states cannot be described as the tensor product of individual qubit states, and that measurements on entangled qubits exhibit correlations that have no classical explanation.

The CNOT gate, which flips the target qubit conditioned on the state of the control qubit, was introduced as the fundamental two-qubit entangling operation. Students learned to apply the CNOT gate in Qiskit circuits and verify its action on all four computational basis states.

The four Bell states were constructed using the standard circuit structure of a Hadamard gate followed by a CNOT gate. Each of the four Bell states was derived by preparing different initial states before applying this circuit. Students executed these circuits on the AerSimulator and observed that measurement outcomes for the two qubits were always perfectly correlated or anti-correlated, confirming entanglement.

The S, S-dagger, T, and T-dagger gates were then studied as phase gate operations that rotate the qubit state about the Z-axis by specific fractions of a full turn. The week concluded with the construction of the three-qubit GHZ state, an entangled state involving three qubits that serves as an important resource in quantum communication and quantum error correction protocols.

WEEK 5 — Introduction to Quantum Algorithms

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Overview of quantum algorithm categories: exact, approximate, sampling, simulation	Understanding the landscape of quantum algorithmic approaches	Faculty
Day 2	Quantum parallelism: evaluating functions on superposition of inputs simultaneously	Understanding the basis of quantum computational advantage	Faculty
Day 3	Phase kickback mechanism: how phase information is transferred from target to control	Learning the key technique used in many quantum algorithms	Faculty
Day 4	Oracle construction: black-box functions encoded as quantum unitary operations	Understanding how computational problems are encoded for quantum algorithms	Faculty
Day 5	Query complexity model: measuring advantage by number of oracle calls	Understanding the framework for proving quantum speedup	Faculty
Day 6	Practical: constructing sample oracle circuits in Qiskit and observing phase effects	Hands-on exploration of oracle design and phase kickback	Faculty

Objective: To introduce the theoretical foundations of quantum algorithm design, including quantum parallelism, phase kickback, oracle construction, and the query complexity framework used to demonstrate quantum advantage.

Detailed Report

The fifth week introduced the conceptual framework underlying quantum algorithm design. Students studied the categories of quantum algorithms, including exact algorithms that solve problems with certainty, approximate algorithms, sampling algorithms, and quantum simulation, understanding the types of problems that each category addresses.

Quantum parallelism was introduced as the principle by which a quantum computer can evaluate a function on all possible inputs simultaneously by applying that function to a superposition state. Students learned why this parallelism does not immediately translate to a speedup for all problem types, as measurement can only reveal one outcome.

The phase kickback mechanism was studied in detail, as it is the key technique used by most of the landmark quantum algorithms covered in the programme. Students learned how a controlled unitary operation on an eigenstate of that unitary transfers phase information from the target register back to the control qubit, enabling the extraction of function properties through interference.

Oracle circuits were introduced as the standard way of encoding classical functions into quantum unitary operations. Students practiced constructing simple oracle circuits in Qiskit and observed the effect of phase kickback on the control register state. The query complexity model was explained as the theoretical framework that measures quantum advantage by comparing the number of oracle queries required by quantum and classical algorithms.

WEEK 6 — Deutsch, Deutsch-Jozsa and Bernstein-Vazirani Algorithms

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Deutsch's Algorithm: problem statement, single-query solution, circuit design	Understanding the first quantum algorithm with provable classical advantage	Faculty
Day 2	Deutsch's Algorithm in Qiskit: implementation, oracle cases, and result verification	Hands-on implementation of Deutsch's Algorithm circuits	Faculty
Day 3	Deutsch-Jozsa Algorithm: generalisation to n-bit functions, circuit structure	Understanding exponential query advantage for constant vs balanced functions	Faculty
Day 4	Deutsch-Jozsa implementation in Qiskit: constant and balanced oracle cases	Implementing and verifying the Deutsch-Jozsa Algorithm	Faculty
Day 5	Bernstein-Vazirani Algorithm: recovering hidden bitstring, Hadamard sandwich circuit	Learning exact hidden string recovery in a single query	Faculty
Day 6	Bernstein-Vazirani in Qiskit: oracle construction for arbitrary secret strings	Implementing and verifying Bernstein-Vazirani for multiple secret strings	Faculty

Objective: To study and implement the Deutsch Algorithm, the Deutsch-Jozsa Algorithm, and the Bernstein-Vazirani Algorithm, understanding the quantum advantage each provides and the circuit structures that realise them.

Detailed Report

The sixth week began with a detailed study of Deutsch's Algorithm, historically significant as the first algorithm to demonstrate a provable quantum advantage over classical computation. Students learned the problem statement: given a function that is either constant or balanced, determine which in a single oracle query. The circuit structure, consisting of a Hadamard gate on the input qubit, the oracle operation, and a final Hadamard gate followed by measurement, was studied and implemented in Qiskit.

The Deutsch-Jozsa Algorithm was then introduced as the generalisation of Deutsch's Algorithm to functions of n-bit inputs. Students learned that while a classical probabilistic algorithm requires O(2^(n-1) + 1) queries to distinguish constant from balanced functions with certainty, the Deutsch-Jozsa Algorithm achieves this in exactly one query. The uniform superposition circuit structure using Hadamard gates on all input qubits was studied and implemented.

The Bernstein-Vazirani Algorithm was then covered as an algorithm for exactly recovering a hidden n-bit string s such that f(x) = s.x mod 2, using a single oracle query. Students observed that the circuit structure is identical to Deutsch-Jozsa, but the measurement of the input register after the final Hadamard layer directly reveals the hidden string.

All three algorithms were implemented in Qiskit with multiple oracle instances, and students verified that the measurement outcomes correctly identified the function type or recovered the hidden string in each case. These algorithms concretely demonstrated how quantum interference enables the extraction of global function properties from a single oracle evaluation.

WEEK 7 — Simon's Algorithm and Grover's Search Algorithm

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Simon's Algorithm: problem statement, hidden XOR structure, exponential classical lower bound	Understanding the exponential separation between classical and quantum query complexity	Faculty
Day 2	Simon's Algorithm circuit: uniform superposition, oracle, and Hadamard layer structure	Learning the circuit design for Simon's Algorithm	Faculty
Day 3	Simon's Algorithm in Qiskit: oracle construction and classical post-processing	Implementing Simon's Algorithm and interpreting measurement results	Faculty
Day 4	Grover's Algorithm: unstructured search problem, amplitude amplification principle	Understanding the quadratic speedup for database search	Faculty
Day 5	Grover oracle and diffusion operator: circuit construction, iteration count	Learning the two-step Grover operator design	Faculty
Day 6	Grover's Algorithm in Qiskit: implementation for various marked states and verification	Hands-on implementation and result analysis of Grover's Algorithm	Faculty

Objective: To study and implement Simon's Algorithm and Grover's Search Algorithm, understanding the quantum speedups they provide and gaining hands-on experience in constructing their respective oracle and diffusion circuits in Qiskit.

Detailed Report

The seventh week commenced with Simon's Algorithm, which solves the problem of finding a hidden XOR mask s such that f(x) = f(x XOR s) for all inputs, using O(n) quantum queries compared to the exponential O(2^(n/2)) queries required classically. Students studied the circuit structure, which applies a Hadamard layer to the input register, evaluates the oracle, and applies a second Hadamard layer before measuring. The measurements yield n-1 linearly independent equations whose solution via classical Gaussian elimination recovers the hidden string.

The implementation of Simon's Algorithm in Qiskit was carried out for specific example oracle functions. Students observed that the measurement outcomes consistently produced vectors orthogonal to the hidden string, as required by the algorithm. The role of classical post-processing in completing the solution was discussed, illustrating how quantum and classical computation are used together.

Grover's Search Algorithm was then introduced as a quantum algorithm providing a quadratic speedup for the problem of finding a marked element in an unstructured database of N items, requiring O(sqrt(N)) oracle queries compared to O(N/2) classically. The amplitude amplification principle was studied: the algorithm iteratively increases the probability amplitude of the marked state while decreasing the amplitudes of the unmarked states.

Students implemented Grover's Algorithm in Qiskit, constructing the oracle that flips the phase of the target state and the Grover diffusion operator that reflects all amplitudes about their average. The circuit was executed for various target states, and students verified that after the correct number of Grover iterations, measurement of the output register produced the marked element with high probability.

WEEK 8 — Final Project, Documentation and Submission

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Final project introduction: QRNG system design, circuit planning, and module breakdown	Understanding the project scope and mapping quantum concepts to implementation	Faculty
Day 2	QRNG circuit implementation: Hadamard gate, measurement, and bit extraction from AerSimulator	Implementing the core quantum random bit generation module	Faculty
Day 3	Number formatting and statistical validation: converting bits to integers, floats, and hex; chi-squared and entropy tests	Completing the output formatting and randomness validation modules	Faculty
Day 4	Applications and security analysis: cryptographic key generation, Monte Carlo estimation, entropy reporting	Demonstrating practical applications of the QRNG system	Faculty
Day 5	Project documentation preparation: writing report sections, code comments, and result analysis	Developing technical writing and documentation skills	Faculty
Day 6	Logbook completion, final review, and internship submission	Finalising all internship deliverables and completing the programme	Faculty

Objective: To design and implement the Quantum Random Number Generator mini project integrating all quantum computing concepts developed during the programme, and to complete all internship documentation including the project report and logbook for final submission.

Detailed Report

The eighth and final week of the internship was dedicated to the development of the final mini project titled Quantum Random Number Generator Using Qiskit. Students began by reviewing the complete project scope and mapping each module to the quantum concepts developed across the preceding seven weeks. The project architecture was planned, identifying the QRNG circuit module, the bit extraction and formatting module, the statistical validation module, and the applications demonstration module.

The core QRNG circuit was implemented using an eight-qubit circuit with Hadamard gates applied to each qubit, creating an equal superposition over all 256 possible eight-bit strings. The circuit was executed on the AerSimulator and measurement outcomes were extracted and reversed to produce a stream of random bits. Students verified that the bit extraction logic correctly converted Qiskit's little-endian measurement strings to MSB-first format.

The number formatting and statistical validation modules were then completed. Students implemented bit-to-integer conversion using binary arithmetic, floating-point normalisation, hexadecimal string formatting, and rejection sampling for range-bounded integers. Statistical tests including the bit frequency test, chi-squared uniformity test, Shannon entropy calculation, and runs test were applied to the generated output to verify its randomness properties.

Application demonstrations were developed, including a 128-bit cryptographic key generator, a quantum-generated passphrase system, a fair dice simulator, and a Monte Carlo estimation of pi using quantum random points. These demonstrations illustrated the practical relevance of true quantum randomness in security and simulation contexts.

The week concluded with the preparation of the project documentation and the completion of the internship logbook. Students reviewed all programme materials, finalised their project reports, and submitted all internship deliverables. The programme concluded with a review session reflecting on the learning journey from quantum computing fundamentals through gate operations and algorithms to the development of a complete quantum application.