Industrial Internship Report on

"Banking Information System"

Prepared by

[Gaurav Pathak]

| |
| :---: |
| *Executive Summary* |
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). |
| This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time. |
| My project was Bank Management System. It's a desktop based application which have similar functionalities like as an Automated teller machine (ATM). |
| This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship. |

TABLE OF CONTENTS

- ## **Preface**

Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned
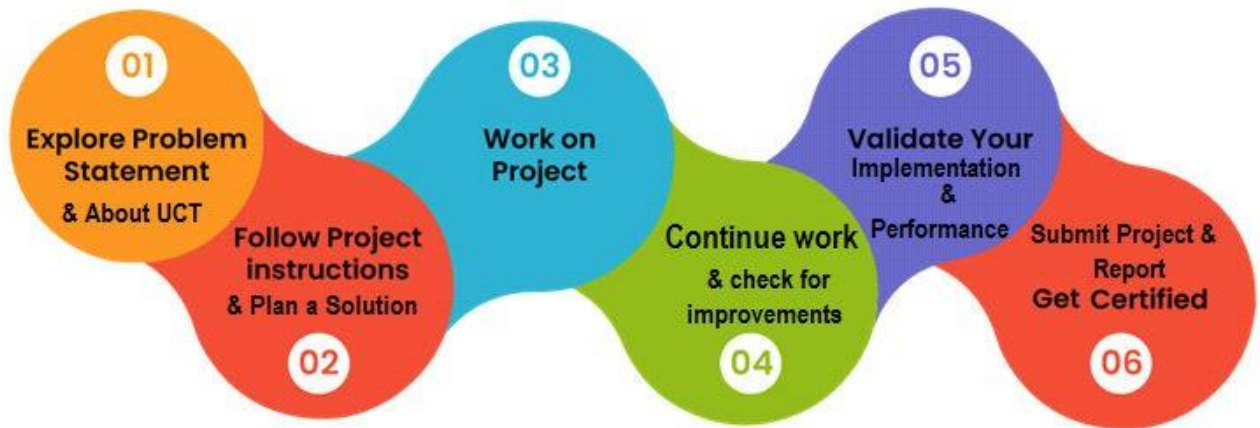
Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

- **Introduction**

- **About UniConverge Technologies Pvt Ltd**

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

- # UCT IoT Platform (*uct* Insight)

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA

- It supports both cloud and on-premises deployments.

It has features to
- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine
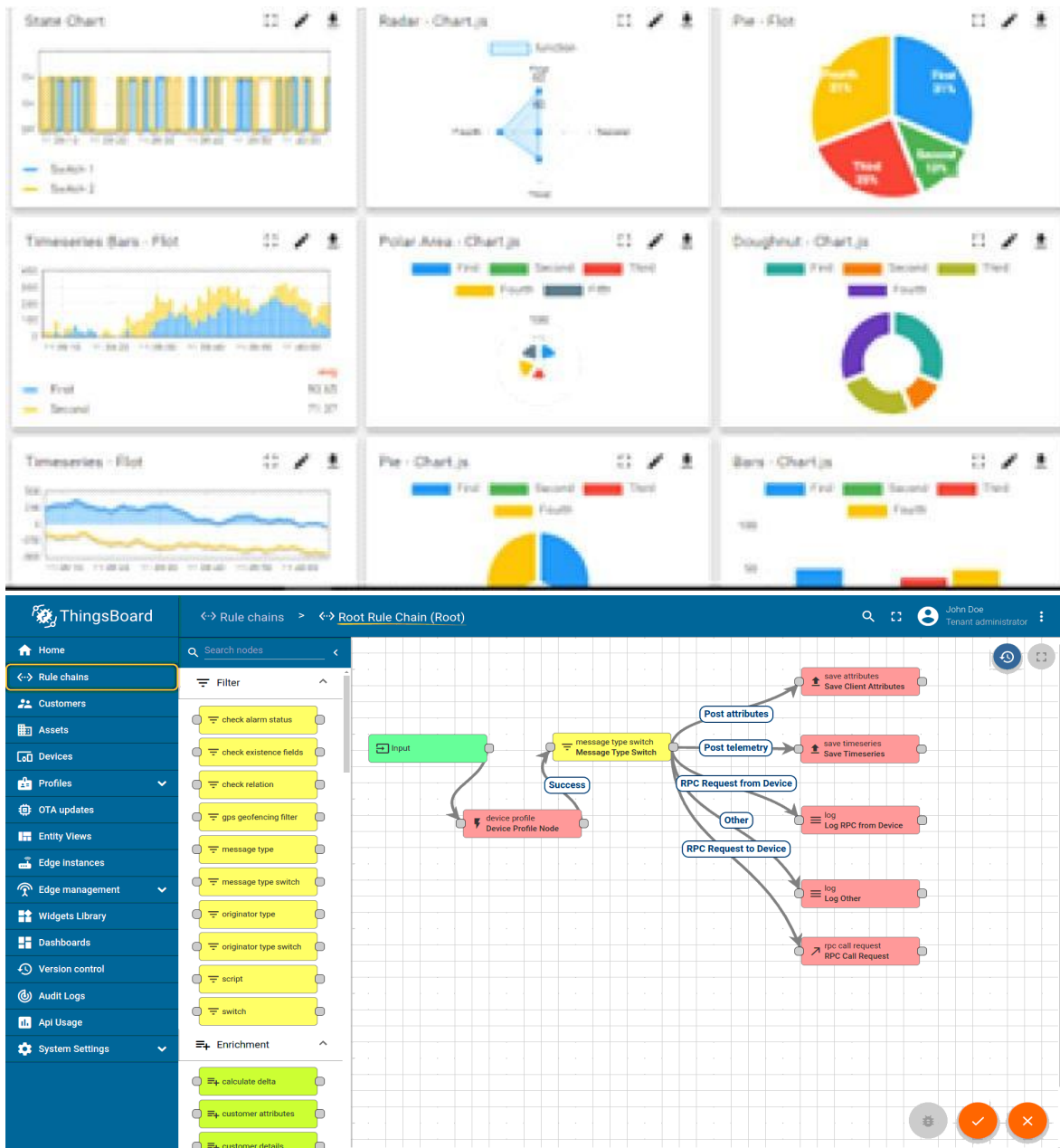
- **Smart Factory Platform (** **)**

Factory watch is a platform for smart factory needs.

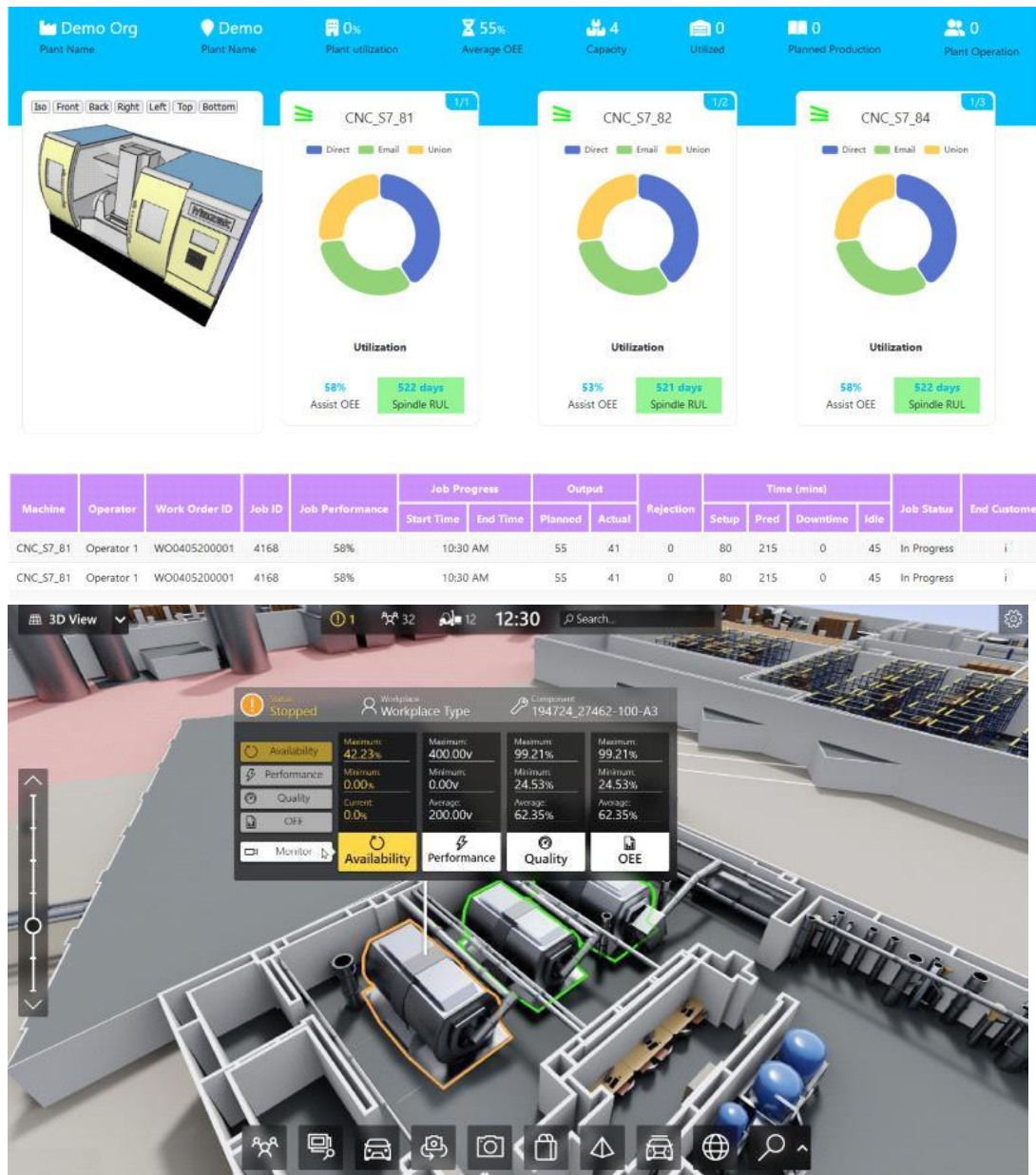It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring

- OEE and predictive maintenance solution scaling up to digital twin for your assets.

- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.

- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

| Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

- **LoRaWAN**™   based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

- ## Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.
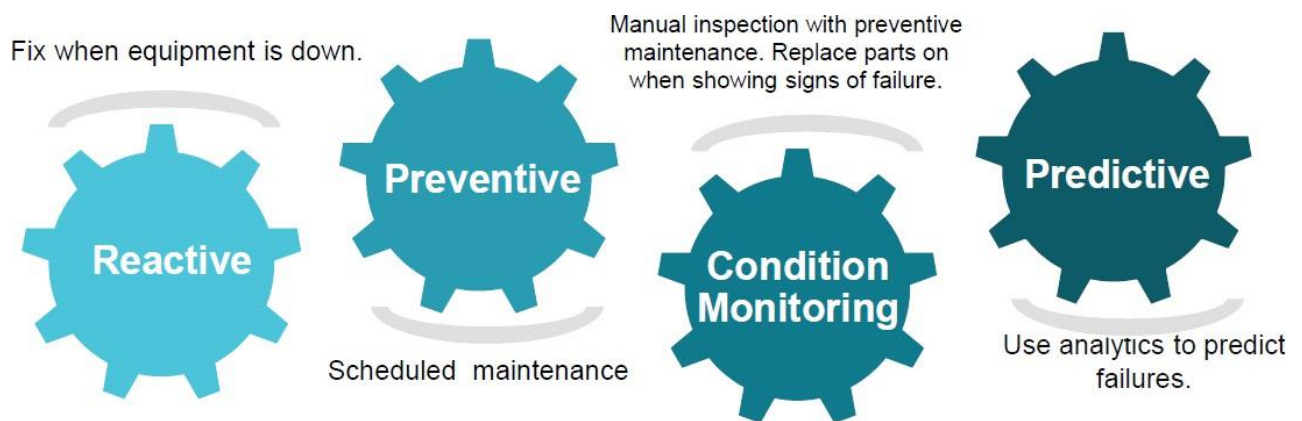

- **About upskill Campus (USC)**

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.

upSkill CAMPUS Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

https://www.upskillcampus.com/
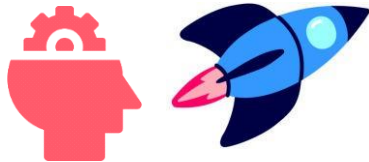
upSkill Campus aiming to upskill 1 million learners in next 5 year\

- **The IoT Academy**

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

- **Objectives of this Internship program**

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

- **Reference**

1. Online Bank Account Management System
Website: http://www.slideshare.net (Collect some info for report documents)

2. PHP and MySQL video tutorials
Website: http://www.freehinditutorial.com, http://www.youtube.com

3. JavaScript form validation: Validate Password, Validate Email, Validate Phone Number, http://webcheatsheet.com/javascript/form_validation.php

- **Glossary**

Terms	Acronym
Internet Banking System	IBS
Application Programming Interface	API
Online Transaction Processing	OLTP
Java Database Connectivity	JDBC
User Interface	UI

- **Problem Statement**

1. Account Management: Provide a platform for users to create and manage their bank accounts efficiently.

2. Transaction Processing: Enable users to deposit and withdraw funds securely and accurately.

3. Transaction History: Maintain a comprehensive record of all user transactions for easy tracking and reference.

4. User Authentication: Implement a secure login system to ensure authorized access to user accounts.

5. Security and Privacy: Incorporate robust security measures to protect user data and prevent unauthorized access.

6. Scalability: Design the system to handle a high volume of concurrent transactions without compromising performance.

The problem statement aims to address the challenges faced in traditional banking systems, such as manual processes, limited access to account information, and potential security vulnerabilities.

• Existing and Proposed solution

Existing solutions for bank account management systems include various software applications and online banking platforms offered by different banks.

Limitations of existing solutions may include limited functionality, lack of user-friendly interfaces, potential security vulnerabilities, and difficulties in customization to specific bank requirements.

Proposed Solution:

- The proposed solution is the development of a Bank Account Management System using Java technologies.

- The system aims to provide a comprehensive set of features for account management, transaction processing, and customer interactions.

- It will include modules for administrative tasks, user operations, and secure access to account information.

Value Addition:

- *Enhanced functionality:* The proposed solution will offer a wide range of banking operations such as account creation, balance maintenance, transaction processing, and fund transfers.

- *Improved user experience:* The system will have an intuitive and user-friendly interface, allowing customers to easily navigate and perform transactions.

- *Robust security measures:* The solution will prioritize data security and implement measures to protect customer information and prevent unauthorized access.

• Code submission (Github link)

Github Link :-   https://github.com/Gaurav-pathak60/upskill-Campus

- **Report submission (Github link) :** https://github.com/Gaurav-pathak60/upskill-Campus

- ## Proposed Design/ Model

The proposed design/model of the Bank Account Management System follows a structured flow from the initial stages to the final outcome.

1. **Requirements Gathering:** The project begins by identifying and understanding the requirements of the system. This involves analyzing the needs of users, stakeholders, and the overall objectives of the bank.

2. **System Design:** Based on the gathered requirements, a system design is created. This includes defining the overall architecture, component interactions, data flow, and user interface design.

3. **Development:** The development phase involves implementing the system design using Java technologies. This includes writing the necessary code, integrating different modules, and ensuring proper functionality.

4. **Testing and Debugging:** Once the development is complete, the system undergoes thorough testing to identify and fix any bugs or issues. Testing involves validating the system's behavior, functionality, and performance to ensure it meets the specified requirements.

5. **Iterative Improvements:** Based on the testing results and user feedback, iterative improvements are made to enhance the system's usability, efficiency, and reliability.

6. **Documentation**: Throughout the design and development process, documentation is created to record the system's architecture, functionality, and usage instructions. This documentation helps in understanding and maintaining the system in the future.

7. **Deployment and Rollout:** Once the system is thoroughly tested and refined, it is deployed in the production environment. This involves setting up the necessary infrastructure, configuring the system, and ensuring its seamless integration with the existing bank infrastructure.

8. **User Training and Support:** Users are provided with appropriate training to understand and effectively utilize the system. Ongoing support is offered to address any user queries or technical issues that may arise.

The final outcome is a fully functional Bank Account Management System that meets the requirements, provides a user-friendly interface, ensures data security, and enables efficient banking operations for customers and bank administrators.

High Level Diagram (if applicable)

• **Low Level Diagram (if applicable)**

• **Interfaces (if applicable)**

Some User Interface Samples of Bank Management System.
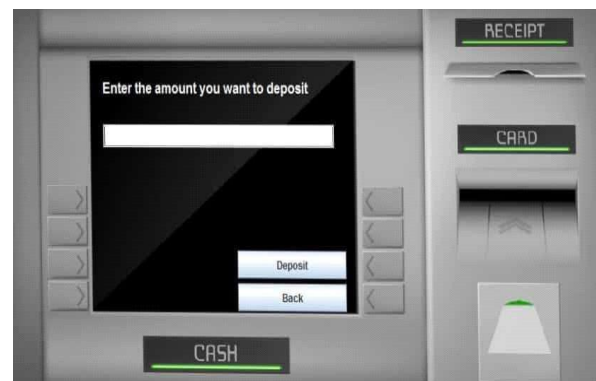
• **Performance Test**

Constraints for Performance of Bank Account Management System:

• *Memory:* Efficient utilization of memory resources, optimized data structures, and regular monitoring to address memory-related issues.

• *Speed and Operations per Second (MIPS):* Design for efficient handling of concurrent transactions, employing efficient algorithms, minimizing complexity, and optimizing code execution.

- <u>Accuracy:</u> Robust data validation, error handling, and comprehensive testing to ensure accurate calculations and transaction handling.

- *<u>Durability:</u>* Handling high data volumes with data integrity and system stability, reliable data storage, transaction logging, and backup/recovery procedures.

- *<u>Power Consumption:</u>* Design for energy efficiency by minimizing resource usage, optimizing code execution, and adopting power-saving techniques.

Importance of Performance Testing:

- Evaluating system performance within identified constraints.
- Assessing scalability, response times, throughput, and limitations.
- Identifying bottlenecks and optimizing for improved performance.
- Recommendations based on test results to enhance scalability and address limitations.

Test Plan/Test Cases :

1. <u>Test Plan Overview :-</u>

- Objectives and scope of testing defined.

- Key functionalities and modules identified.

- Testing approach, environment, and resources specified.

2. <u>Test Cases :-</u>

Account Creation:

- Test case 1: Verify successful creation of a new account with valid user information.

- Test case 2: Verify error display for invalid or missing information during account creation.

Deposit and Withdrawal:

- Test case 3: Verify accurate updating of account balance upon deposit.

- Test case 4: Verify correct deduction from account balance upon withdrawal.

Transaction History:

- Test case 5: Verify accurate display of transaction details in history.

- Test case 6: Verify correct timestamps and transaction amounts in history.

User Authentication:

- Test case 7: Verify successful login and access to account information with valid credentials.

- Test case 8: Verify error message display for invalid login attempts.

Data Validation:

- Test case 9: Verify proper validation of user inputs to prevent unauthorized actions or data corruption.

- Test case 10: Verify appropriate error messages for invalid data entries.

3. <u>Test Procedure:</u>

- Execute test cases following the testing approach.

- Record actual results, noting discrepancies or issues encountered.

- Document bug reporting and resolution steps.

- Repeat test cycles as needed for comprehensive coverage.

Performance Outcome:

- Measure system performance based on predefined metrics (response time, throughput, resource usage).

- Utilize suitable tools (load testing, stress testing) to simulate real-world scenarios.

- Analyze results to identify bottlenecks, limitations, and areas for improvement.

- Evaluate if system meets performance requirements defined in the test plan.

Note: Adapt the test plan, test cases, procedure, and performance outcome to fit the Bank Account Management System's specific requirements and functionalities.

My learnings

Working on the Bank Account Management System project has provided valuable learnings for my career growth:

1. Enhanced technical skills in Java, SQL, and object-oriented programming.
2. Improved project management abilities through requirements gathering and system design.
3. Strengthened problem-solving skills by overcoming implementation challenges.
4. Developed effective collaboration and communication skills within a team.
5. Gained industry-specific knowledge in the banking domain.

These learnings will empower me to excel in future projects, tackle complex tasks, and contribute effectively to the technology industry.

Future work scope

1. Enhanced security features: Implement two-factor authentication, encryption, and integration with secure payment gateways.

2. Advanced reporting and analytics: Incorporate data visualization and analytics for comprehensive insights into customer transactions and financial performance.

3. Integration with online banking services: Enable transactions, balance checks, and account access through web and mobile interfaces.

4. Integration with third-party services: Partner with payment gateways, credit bureaus, and identity verification providers to enhance system functionality.

5. Mobile application development: Create a mobile app for seamless banking experiences on smartphones.

6. Automated transaction categorization: Use machine learning to categorize transactions and provide spending insights.

7. System scalability: Optimize architecture for increased user and transaction volume.

8. User experience enhancements: Continuously improve the interface, navigation, and overall user experience.

These future work scopes allow for further development and refinement of the Bank Account Management System to meet evolving user needs and industry standards.

In conclusion, this project has been a valuable learning experience, equipping me with skills for career growth. The Bank Account Management System is expected to meet users' banking needs by providing a reliable and efficient account management platform. I am excited to apply my learnings and contribute to future finance and technology projects.

[Thank You]