

# **AI-Driven Product Design Internship**

*by*

**Masfiq Mahmud (212014007)**

*Internship report (CSE4099) submitted in partial fulfillment of the  
requirements for the degree of*

**Bachelor of Science in Computer Science and Engineering**

Under the supervision of

**Nasir Uddin Ahmed**

Lecturer

**Md Asib Rahman**

Technical Lead



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UNIVERSITY OF LIBERAL ARTS BANGLADESH**

**FALL 2023**

© *Author*  
All rights reserved

# DECLARATION

|                              |                                 |
|------------------------------|---------------------------------|
| <b>Internship Title</b>      | AI-Driven Product Design Intern |
| <b>Author</b>                | <i>Masfiq Mahmud</i>            |
| <b>Student ID</b>            | 212014007                       |
| <b>Supervisor (Academic)</b> | Nasir Uddin Ahmed               |
| <b>Supervisor (Industry)</b> | Md. Asib Rahman                 |

---

I declare that this internship report entitled *AI-Driven Product Design Intern* is the result of my own work except as cited in the references. The internship report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

---

**Masfiq Mahmud**  
**212014007**

Department of Computer Science and Engineering  
University of Liberal Arts Bangladesh

**Date: August 28, 2025**



Department of Computer Science and Engineering  
University of Liberal Arts Bangladesh  
Mohammadpur, Dhaka - 1207

---

## CERTIFICATE

This is to certify that the internship report entitled **Title of Internship**, submitted by **Masfiq Mahmud** (Student ID: 212014007) is a undergraduate student of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

---

**Nasir Uddin Ahmed, Lecturer**  
University of Liberal Arts Bangladesh

---

**Md. Asib Rahman, Sr. AI Engineer**  
Robofication LLC.

**Place:** Dhaka

**Date:** August 28, 2025

## ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my supervisor, , for giving me the opportunity to conduct this internship and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance. Also, I would like to thank the other employees of *Robofication LLC* for their help and cooperation during my practice time.

I am greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh who taught me during the course of our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

I am extremely thankful to my parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing me for the future. I would like to say thanks to my friends and relatives for their kind support and care.

Finally, I would like to thank all the people who have supported me to complete the Internship directly or indirectly.

*Masfiq Mahmud*

**University of Liberal Arts Bangladesh**

**Date:** August 28, 2025

Dedicated to  
*my loving Mother*

– *Masfiq Mahmud*

## EXECUTIVE SUMMARY

This report details my internship as an AI-Driven Product Design Intern at Robofication LLC., undertaken to bridge the gap between academic theory and real-world software engineering practice. My primary objective was to immerse myself in a professional development environment and contribute to live projects. Working within the backend development team, my role was governed by the company's Agile methodology, with work structured into two-week sprints. The scope of my responsibilities evolved from initial observation to active implementation of features and critical bug fixes.

My most significant achievement was identifying and resolving a production bug that caused user verification emails to be sent to spam. This fix directly improved the user onboarding process and had a tangible, positive impact on the company's user acquisition goals. I also successfully developed and merged several new features, contributing directly to the value of the core product.

The primary challenge was adapting to the industry's high standards for code quality and collaborative workflows, a stark contrast to solo academic projects. This was overcome through invaluable mentorship and a rigorous code review process, which taught me to apply best practices like SOLID and DRY principles. Through this experience, I gained practical proficiency in key technologies including Node.js, Git, GitHub, and Jira. The internship was a transformative experience, evolving my skills from basic programming to professional software craftsmanship and instilling the discipline and confidence needed for a career in software engineering.

**Keywords:** Agile Methodology, Code Review, Git Workflow, Software Engineering

# Contents

|          |                                                        |           |
|----------|--------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                    | <b>1</b>  |
| 1.1      | Introduction . . . . .                                 | 1         |
| 1.2      | Objective of the internship report . . . . .           | 1         |
| <b>2</b> | <b>Background</b>                                      | <b>3</b>  |
| <b>3</b> | <b>Scope of Work</b>                                   | <b>5</b>  |
| <b>4</b> | <b>Internship Activities</b>                           | <b>6</b>  |
| 4.1      | Introduction of the Organization/Institution . . . . . | 6         |
| 4.2      | Work Accomplishment . . . . .                          | 7         |
| 4.3      | Challenges Faced . . . . .                             | 9         |
| <b>5</b> | <b>Methodology</b>                                     | <b>10</b> |
| 5.1      | Methodology . . . . .                                  | 10        |
| <b>6</b> | <b>Analysis and Findings</b>                           | <b>12</b> |
| 6.1      | Analysis and Findings . . . . .                        | 12        |
| 6.1.1    | Results/Findings . . . . .                             | 12        |
| 6.1.2    | Learning and Development . . . . .                     | 13        |
| 6.1.3    | Tools and Technology . . . . .                         | 14        |
| 6.1.4    | User Guideline . . . . .                               | 14        |
| 6.1.5    | Impact of the Internship . . . . .                     | 15        |
| <b>7</b> | <b>Conclusions and Recommendations</b>                 | <b>16</b> |
| 7.1      | Conclusion . . . . .                                   | 16        |
| 7.2      | Limitations . . . . .                                  | 17        |
| 7.3      | Recommendations . . . . .                              | 17        |

|                          |           |
|--------------------------|-----------|
| <b>8 Acknowledgments</b> | <b>19</b> |
| <b>A Offer Letter</b>    | <b>20</b> |
| <b>References</b>        | <b>22</b> |

# Chapter 1

## Introduction

### 1.1 Introduction

As an AI Intern at Robofication LLC.ai, I am immersed in pioneering work that leverages state of the art Large Language Models to democratize the creation and comprehension of Systems Modeling Language (SysML) diagrams for mission critical industries. By translating natural language requirements into rigorous, executable system models, our platform is poised to accelerate design cycles in automotive ECU networks and software defined vehicles, streamline avionics integration for aerospace OEMs, enhance patient safety modeling in medical devices, optimize renewable energy grid architectures. This report chronicles my contributions to refining the LLM pipeline spanning prompt engineering, AI Agent and system validation toward a future where complex system design becomes as intuitive as conversing with an expert colleague.

### 1.2 Objective of the internship report

Robofication LLC., spun out from RoboFication LLC in April 2025, is an AI-native platform whose mission is to eliminate the most stubborn bottleneck in safety critical engineering: manual modeling. Guided by the vision that “engineers don’t just draw models they collaborate with AI to design and validate systems faster, safer, and smarter,” Robofication LLC converts plain English, code, or voice into fully compliant SysML diagrams in minutes instead of weeks, initially targeting automotive ECU networks, aerospace flight control stacks, and defense mission

systems. The company's pedigree rooted in 15+ years of ISO 26262 and formal methods leadership at Ford, GM RD, MIT, and TUM underpins a roadmap that begins with AI-powered modeling and modularly expands into requirements, analysis, and verification. The purpose of my internship was to accelerate this roadmap by serving as an AI Intern bridging backend, frontend, and prompt-engineering efforts: I architected agentic features that orchestrate LLMs to autonomously refine and verify SysML models, implemented domain-aware email classification (e.g., .edu vs. corporate) to streamline customer onboarding and support, and iteratively tuned prompts to boost model accuracy and safety compliance. This report documents the technical, product, and strategic insights gained while translating Robofication LLC's cutting edge research into production-ready capabilities that democratize rigorous systems engineering across safety critical industries.

# Chapter 2

## Background

Robofication LLC. was incorporated in April 2025 as a focused spin out from RoboFication LLC, a research group that pioneered the concept of an “AI Agent as Systems Engineer.” While RoboFication pursued end-to-end automation of the entire systems-engineering lifecycle, Robofication LLC now serves as the core wedge an AI-native platform that converts plain English, source code, or voice input into fully compliant SysML diagrams in minutes rather than weeks. The company’s mission is to eliminate manual modeling as the fundamental bottleneck in safety-critical product development; its long-term vision is a collaborative loop in which engineers and AI agents co-design, analyze, and verify complex systems faster, safer, and smarter. Guided by the founder’s 15-year track record at Ford (ISO 26262-compliant processes, autonomous-system safety, formal verification) and research collaborations with GM RD, MIT, University of Michigan, and TUM, Robofication LLC initially targets automotive ECU networks, aerospace flight-control architectures, and defense mission systems industries where rigorous model-based engineering is mandatory yet painfully slow.

The purpose of my internship was to accelerate the productization of this vision. As an AI Intern I operated across the full stack: on the backend I built agentic orchestration workflows that allow large language models to iteratively refine, simulate, and formally verify SysML artifacts; on the frontend I integrated these agents into an intuitive web IDE so users can converse with the AI in real time; and across both layers I engineered domain-aware email classification (distinguishing .edu, corporate, and government domains) to automate customer onboarding and support triage. Continuous prompt-engineering experiments were used to enhance accuracy. This report captures the technical, product, and strategic insights gained

while turning cutting edge research into production features that bring rigorous systems modeling within reach of every safety-critical engineering team.

# Chapter 3

## Scope of Work

During my internship at **Robofication LLC.**, my responsibilities were focused on contributing to the core functionality of the AI platform. I was tasked with projects that spanned backend development, frontend integration, and direct enhancement of the Large Language Model's performance. The specific tasks assigned to me were as follows:

- **Development of an Agentic Feature:** My primary project involved developing a new "agentic feature" for the platform.
  - **Backend Development:** Built the backend logic enabling the LLM to autonomously handle user requests for SysML diagram generation.
  - **Frontend Integration:** Integrated these backend capabilities into the user-facing application, ensuring accessibility and functionality for end-users.
- **Implementation of Domain Email Classification:** Created a module to classify user emails by domain type (e.g., '.edu', corporate, etc.) to improve internal workflows for user management and support.
- **LLM Output Enhancement via Prompt Engineering:** Continuously improved SysML diagram validation quality through systematic prompt engineering, ensuring outputs aligned with user intent and industry standards.

# Chapter 4

## Internship Activities

### 4.1 Introduction of the Organization/Institution

I completed my internship at Robofication LLC., a forward-thinking technology company operating at the intersection of software engineering and product innovation.

Robofication LLC. operates on a highly disciplined and efficient Agile methodology, which forms the backbone of its operational rhythm. The development cycle is structured into two-week sprints, creating a fast-paced, results-oriented environment that culminates in a production release every other Thursday. A cornerstone of this workflow is the daily stand-up meeting, a brief but critical session that ensures constant team alignment, transparency, and rapid problem-solving. Project management is centralized through Jira, which is used to track all tasks, manage sprints, and ensure accountability across the team. This structured approach fosters a culture of discipline and continuous momentum.

The organization places a profound emphasis on software quality and collaborative craftsmanship. The development process is standardized around a robust Git and GitHub workflow, where every contribution is made on a dedicated branch and subjected to a formal Pull Request (PR) and peer review process. This practice is not merely a technical step but a cultural one, where senior developers actively mentor junior members by providing constructive feedback on their code. This rigorous review ensures that all code adheres to industry best practices, such as the SOLID and DRY principles, promoting a codebase that is clean, maintainable, and scalable.

## 4.2 Work Accomplishment

### Week 1 - 2: Immersion into the Agile Workflow and Company Processes

- **Observed & Experienced:** I was immersed in the company's operational rhythm, based on a two-week sprint cycle that culminates in a production release every other Thursday.
- **Learned:** I participated in daily stand-up meetings, which are efficient 1-to-1:30-hour sessions for sharing progress, plans, and blockers. I learned that this regular communication is vital for maintaining momentum and ensuring team alignment.
- **Gap Identification:** The structured daily stand-up contrasted sharply with the often infrequent and unstructured meetings in university projects. This highlighted the importance of disciplined, regular communication in a professional setting.
- **Learned & Understood:** I was introduced to Jira for project management. Learning to navigate the board and track tasks provided hands-on experience with an industry-standard tool, a practical application missing from my theoretical project management studies.

### Week 3 - 6: Mentorship, Code Quality, and First Contributions

- **Performed:** Under my mentor's guidance, I began active contribution by implementing specific tasks, such as creating an API endpoint within a 30-minute deadline, which taught me to code with speed and focus.
- **Learned:** The code review process was instrumental to my growth. I learned crucial best practices like the SOLID and DRY (Don't Repeat Yourself) principles, which were only superficially covered in my coursework.
- **Gap Identification:** I discovered a significant gap between academic and professional coding. While university focused on code that produced a correct output, the organization required code that was also clean, readable, efficient,

and maintainable by a team. I understood that writing “good” code is a collaborative discipline.

## **Week 6 - 10: Mastering the Collaborative Development Workflow**

- **Learned & Performed:** I became proficient in the company’s Git and GitHub workflow. I learned to create a dedicated, professionally named Git branch for every new feature or bug fix.
- **Experienced:** I mastered the cycle of creating a Pull Request (PR), receiving collaborative feedback from senior developers, and making iterative changes until my code was approved and merged.
- **Gap Identification:** This workflow revealed a gap in my education, which had not prepared me for the collaborative nature of software development. I understood that version control in a professional team is the backbone for collaboration, quality assurance, and maintaining codebase integrity, not just a tool for personal backups.

## **Week 11 and Subsequent Weeks: Taking Ownership and Delivering Impact**

- **Performed:** I was entrusted with greater responsibility, moving from small tasks to fixing real user-impacting bugs and developing new features that were successfully merged into production.
- **Accomplishment:** My most significant achievement was resolving a critical bug that caused user verification emails to go to spam. I investigated the root cause, implemented a fix for the email headers, and successfully deployed it, ensuring emails reached user inboxes.
- **Learned:** I learned a systematic approach to debugging complex, real-world problems, which required research and a holistic system understanding. On-the-job learning of technologies like Node.js, not covered in my curriculum, was essential.

- **Gap Identification:** This task highlighted the gap between solving well-defined academic problems and tackling messy, multifaceted issues in a live production environment. I understood the direct and tangible impact my work could have on the business and its customers.

## 4.3 Challenges Faced

My internship at **Robofication LLC** provided a valuable learning curve, presenting several technical and operational challenges that were instrumental to my professional growth. The key obstacles I encountered and overcame included:

- **Transition to Professional Development Environment:**
  - Shifted from academic projects to collaborative development using industry tools (Git/GitHub)
  - Learned team-specific workflows, branching strategies, and pull request protocols through focused practice
- **Onboarding Complex Codebase:**
  - Navigated a large, sophisticated codebase with intricate architecture
  - Dedicated time to study module interactions and identify implementation areas

### Resolution Strategies:

- Leveraged Robofication LLC's collaborative culture by proactively seeking guidance
- Received consistent mentorship from team members through:
  - Technical question sessions
    - \* Code reviews
    - \* Participation in team discussions
- Accelerated learning through observation of established practices

These experiences ultimately enabled me to contribute effectively to my assigned projects while developing crucial professional competencies.

# Chapter 5

## Methodology

### 5.1 Methodology

My approach during the internship was a blend of established industry methodologies and hands-on, problem-driven learning. The work was not conducted in a vacuum but was guided by a structured framework that emphasized collaboration, quality, and continuous improvement. The key methods employed are detailed below.

- **Agile and Scrum Framework:** The entire workflow was anchored in the Agile methodology, specifically using a Scrum-like structure. Work was organized into two-week sprints, which provided a predictable rhythm for development and delivery. The primary method for daily progress tracking and data collection on team status was the daily stand-up meeting. This structured session required me to synthesize my previous day's work, plan my current day, and identify any impediments, ensuring constant alignment with team goals. Project progress, task allocation, and effort estimation were managed using a Jira board, which served as the central repository for all work-related data.
- **Structured Version Control and Peer Review:** The technical development process followed a strict, collaborative methodology centered on Git and GitHub. The approach for any new task (feature or bug fix) was as follows:
  1. Create a new, descriptively named Git branch to isolate changes.
  2. Implement the required code changes on this branch.

3. Submit a Pull Request (PR) on GitHub to propose merging the changes.
4. Engage in a formal code review process, where my mentor and senior developers would analyze the code and provide feedback.
5. Iteratively refine the code based on the feedback received.

This peer-review cycle was a critical quality assurance method, ensuring that all code adhered to best practices like SOLID and DRY principles before being integrated into the main codebase.

- **Mentorship-Driven Skill Development:** A core part of my methodology for learning and growth was direct mentorship. My industry supervisor would assign well-defined, incremental tasks. This approach allowed me to build skills progressively. The primary learning method was the iterative feedback loop established through code reviews. By receiving specific, actionable critiques and performing revisions, I was able to actively bridge the gap between theoretical knowledge and industry-standard practices.
- **Investigative Problem-Solving and Research:** For complex tasks, particularly bug-fixing, my approach was to conduct systematic research and root cause analysis. When tackling the issue of user verification emails landing in spam, for example, my data collection method involved a multi-pronged investigation:
  - Observing the system's behavior and replicating the issue.
  - Analyzing the technical data, specifically the headers and configuration of the outgoing emails.
  - Conducting external research on email deliverability best practices and potential server configuration issues.

This method of systematic investigation, followed by hypothesis testing and implementation of a fix, proved essential for resolving real-world production issues that extended beyond simple code logic.

# Chapter 6

## Analysis and Findings

### 6.1 Analysis and Findings

#### 6.1.1 Results/Findings

My internship yielded several tangible results and key findings, both in terms of direct contributions to the company's product and in understanding the dynamics of a professional software development environment.

- **Resolution of a Critical Production Bug:** My most significant technical result was identifying and fixing a critical issue where user verification emails were being sent to spam folders. The successful resolution directly improved the user onboarding experience, removed a barrier to new user registration, and helped protect the company's professional reputation.
- **Contribution to Production Codebase:** I successfully developed several new features and bug fixes, which were approved through the rigorous code review process and merged into the main production codebase. This demonstrates my ability to produce code that meets industry standards for quality, maintainability, and functionality.
- **Proficiency in Agile Workflow:** A key finding was my rapid and successful integration into the team's Agile workflow. I became proficient in participating in daily stand-ups, managing tasks on the Jira board, and contributing effectively within the two-week sprint cycle.

- **Identification of Theory-Practice Gap:** A recurring finding throughout the internship was the significant gap between academic computer science education and the practical, collaborative nature of industry software engineering. Key areas of discrepancy included the emphasis on code quality (SOLID, DRY), the necessity of collaborative version control (Git workflow), and the use of industry-standard project management tools.

### 6.1.2 Learning and Development

The internship was a period of intense learning and professional development, providing lessons that extended far beyond the technical skills learned in university.

- **Collaborative Software Craftsmanship:** The most profound lesson was that professional software development is a collaborative craft. I learned that writing functional code is only the first step; writing clean, maintainable, and efficient code that can be understood and extended by a team is paramount. The code review process was the primary vehicle for this lesson.
- **Importance of Process and Discipline:** I learned that disciplined processes are the bedrock of an effective development team. The daily stand-ups, structured sprints, and formal Pull Request workflow are not bureaucratic overhead but essential mechanisms for ensuring alignment, maintaining momentum, and guaranteeing quality.
- **On-the-Job Technical Skill Acquisition:** I gained practical, hands-on experience with modern technologies not covered in my curriculum, most notably **Node.js**. I learned how to apply these technologies to solve real-world problems, such as implementing API endpoints and debugging complex system interactions.
- **Problem-Solving in a Live Environment:** I learned to approach problems systematically in a high-stakes production environment. This involves not just coding, but also research, root cause analysis, and understanding the broader business impact of a technical issue.

### 6.1.3 Tools and Technology

During my internship, I utilized a range of industry-standard tools and technologies that were integral to the development workflow.

- **Jira:** Used as the primary project management tool for tracking tasks, managing sprints, and monitoring progress. It served as the central hub for understanding team priorities and individual responsibilities.
- **Git:** The distributed version control system used for tracking changes in the source code. I used it daily for creating branches, committing changes, and managing my local development history.
- **GitHub:** The web-based platform for hosting the Git repositories. It was the central tool for collaboration, primarily used for creating Pull Requests, conducting code reviews, and merging approved code into the main branch.

### 6.1.4 User Guideline

While my work focused on backend features and bug fixes rather than a single user-facing application, the following guidelines describe how to interact with the systems I worked on from both a user and developer perspective.

- **For End-Users (User Verification Process):**
  1. After signing up for a new account, you will receive a verification email.
  2. Following the fix I implemented, this email should now arrive directly in your primary inbox. Please check there first.
  3. If the email is not in your inbox within a few minutes, check your spam or junk folder as a secondary measure.
  4. Click the verification link within the email to activate your account.
- **For Developers (Contributing via Git Workflow):**
  1. Before starting a new task, pull the latest changes from the main development branch.
  2. Create a new, dedicated branch for your feature or bug fix using a descriptive name (e.g., 'feature/add-user-profile' or 'fix/email-header-bug').

3. Implement your changes and commit them with clear, concise messages.
4. Push your branch to GitHub and create a Pull Request (PR) targeting the main development branch.
5. Assign a reviewer (e.g., your mentor or a senior developer) and wait for feedback. Address all comments and push subsequent changes to the same branch until the PR is approved and merged.

### 6.1.5 Impact of the Internship

My contributions during the internship had a direct and positive impact on the organization's projects and operational goals.

- **Enhanced User Experience and Growth:** By fixing the critical email deliverability bug, I directly contributed to the company's growth. The fix removed a significant friction point in the user acquisition funnel, ensuring new users could successfully verify their accounts. This improved the overall user experience and protected the company's brand image.
- **Increased Product Value:** Through the development of new features and the successful merging of my code into production, I directly added value to the company's core product. My work expanded the software's capabilities and contributed to the team's ability to meet its sprint goals and product roadmap objectives.
- **Strengthened Team Velocity:** By rapidly adapting to the company's Agile and Git workflows, I became a productive member of the development team. My ability to take ownership of tasks, from small implementations to significant bug fixes, helped increase the team's overall capacity and velocity, allowing senior members to focus on more complex architectural challenges.

# Chapter 7

## Conclusions and Recommendations

### 7.1 Conclusion

My internship at Robofication LLC. was a transformative experience that successfully bridged the gap between the theoretical knowledge acquired in academia and the practical, dynamic demands of the software engineering industry. Through hands-on experience, I moved from a student's understanding of programming to a professional's appreciation for software craftsmanship. The immersion in an Agile environment, governed by two-week sprints and daily stand-ups, instilled a discipline and collaborative spirit that cannot be taught in a classroom.

The core of my development came from the mentorship-driven approach and the rigorous code review process. I learned that functional code is merely the baseline; the industry standard demands code that is clean, maintainable, and scalable, guided by principles like SOLID and DRY. Mastering the company's Git and GitHub workflow was a pivotal achievement, transitioning my view of version control from a personal backup tool to the essential backbone of team collaboration and quality assurance.

My most significant contribution, fixing the critical user verification email bug, was a powerful lesson in the real-world impact of a developer's work. It solidified my problem-solving skills and demonstrated my ability to deliver tangible value to the business. Ultimately, this internship has equipped me with a robust set of technical skills in tools like Node.js and Jira, a professional methodology for collaborative development, and the confidence to begin a successful career in software engineering.

## 7.2 Limitations

While the internship provided comprehensive and in-depth experience, it was subject to certain inherent limitations:

- **Time Constraint:** The duration of the internship naturally limited my exposure to the full lifecycle of a long-term project. I was involved in specific sprints and features but did not experience the initial project planning, architectural design.
- **Scope of Work:** As an intern, my tasks were understandably well-defined and focused primarily on backend feature implementation and bug fixing. This limited my exposure to other critical areas of the business, such as high-level product strategy, DevOps, or direct client interaction.
- **Initial Learning Curve:** My initial lack of familiarity with industry-specific tools like Jira and professional workflows like the structured Pull Request process presented a limitation at the outset. While this was overcome through mentorship and hands-on work, it required an initial period of adaptation focused on learning rather than maximum productivity.

## 7.3 Recommendations

Based on my positive and enriching experience, I would like to offer the following recommendations to the organization to further enhance its excellent internship program:

- **Develop a Formalized Intern Onboarding Guide:** While the on-the-job mentorship was superb, a concise onboarding document could accelerate an intern's initial integration. This guide could include a brief overview of the core architecture, a step-by-step guide to the Git workflow (branch naming, PR templates), and quick-start instructions for setting up the local development environment. This would empower future interns to contribute even faster.
- **Introduce Structured Knowledge-Sharing Sessions:** I would recommend implementing short, informal weekly or bi-weekly tech talks (15-30 minutes) for interns. A senior developer could present on a specific topic relevant to

the company's stack or methodology, such as a deep dive into the SOLID principles in practice, an overview of the system architecture, or a tutorial on an advanced Git feature. This would formalize the transfer of crucial knowledge that is often learned ad-hoc.

- **Facilitate Broader Exposure through Shadowing:** To help interns understand the full product lifecycle, consider offering a half-day shadowing opportunity. Allowing an intern to shadow a member of the frontend, QA, or DevOps team could provide invaluable context and a more holistic understanding of how their work fits into the larger organizational goals.

## Chapter 8

### Acknowledgments

I would like to take advantage of this opportunity and especially thank our CEO and founder Md Tawhid Bin Waez for granting me this internship opportunity. He took me into the company during my time of need. I will always be grateful for this learning experience that I have had because of him. From the bottom of my heart I pray for his success and for company's bright future as well.

(1? )

# Appendix A

## Offer Letter

---

### Robofication LLC.

House: 210 New Elephant Rd, Dhaka 1205

Website: [sysmodeler.ai](http://sysmodeler.ai)

---

**Date:** June 1, 2025

Dear Masfiq Mahmud,

Email: [masfiqmahmud56@gmail.com](mailto:masfiqmahmud56@gmail.com)

Phone: (+88) 01876892094

We are pleased to offer you the position of **AI Intern** at **RoboFication LLC**. Please find the terms and conditions of your role below:

#### Position Details

- **Position:** AI Intern
- **Working Hours:** 2:00 PM – 11:00 PM, Sunday to Thursday
- **Duration:** 3 months
- **Start Date:** June 1, 2025

#### Notice Period

- Employees must provide 10 working days' notice before resignation.
- Employment may also be terminated immediately.

#### Confidentiality and Data Security

- You must not share any proprietary or confidential information of RoboFication LLC with third parties.
- All RoboFication LLC-related data must be deleted by your last day of employment.
- Any violations may lead to immediate termination and may be reported to:
  - The police
  - Bangladesh Association of Software and Information Services (BASIS)
  - Other relevant authorities

**Exclusivity Clause**

- You may not work for another company (full-time, part-time, volunteer, or contract-based) without prior written permission from RoboFication's management.
- Violations may result in immediate termination and legal action.

**Professional Conduct**

- Demonstrate professionalism, integrity, and dedication.
- Respect all colleagues, regardless of position, gender, origin, or religion.
- Avoid political discussions in the workplace.
- Use English for all work-related communication.

**Required Documents**

Please submit:

- A copy of your National ID card
- Your permanent and current address
- One passport-size photo
- Contact details of two recommenders (full names, phone numbers, and addresses)
- The phone number of someone who lives with you
- Your university transcript

We're excited to have you join the team and contribute to RoboFication's groundbreaking projects.

Md Tawhid Bin Waez, PhD  
CEO & Founder, RoboFication LLC

# References

Robofication. <https://robofication.com/>.