CHAPTER 1

COMPANY INFORMATION

1.1 Profile

HashedIn by Deloitte is a leading product development and technology consulting firm that specializes in building intelligent digital solutions using cloud-native technologies. Headquartered in India and operating globally, HashedIn was founded with a vision to deliver scalable, robust, and high-impact software products that transform the way businesses operate. In 2020, it became a part of **Deloitte**, one of the "Big Four" global professional services firms, further strengthening its capabilities and global reach.

HashedIn is widely recognized for its excellence in software engineering, with a strong emphasis on **Agile development**, **DevOps practices**, and modern software architectures such as **microservices**, **event-driven systems**, and **API-first development**. It caters to a wide range of clients including Fortune 500 companies, fast-growing startups, and mid-size enterprises, primarily in domains such as healthcare, fintech, retail, and SaaS.

The company promotes a **developer-first culture**, encouraging innovation, learning, and ownership at every level. Teams at HashedIn follow modern engineering practices including **continuous integration and continuous deployment (CI/CD)**, **test-driven development**, and **cloud-native application design** using platforms like AWS, GCP, and Azure.

HashedIn’s acquisition by Deloitte enabled it to integrate its lean startup-style product development expertise with Deloitte’s deep industry knowledge, global delivery capabilities, and strategic consulting services. This unique combination allows HashedIn to deliver end-to-end digital transformation services—from consulting and architecture to development, testing, deployment, and scaling.

Through programs like the **HU University Program**, HashedIn also invests significantly in nurturing early talent, offering hands-on experience in full-stack development, AI-driven automation, cloud infrastructure, and enterprise-grade security practice

1.2 Description of the Business Unit/Team

I was part of the **HU University Program**, an intensive internship bootcamp organized by **HashedIn by Deloitte**. This program was designed to bridge the gap between academic knowledge and industry expectations by providing hands-on exposure to modern software development practices. Over a span of six weeks, we followed a sprint-based structure that emphasized practical application, collaboration, and iterative learning.

The curriculum covered a wide range of topics including **full-stack web development**, **DevOps**, **cloud infrastructure**, **software security**, and **generative AI**. From the outset, we worked in an Agile environment that simulated real-world project cycles. Each week focused on different technologies, helping us develop both depth and breadth in our technical skills.

As a **Trainee Engineer**, I contributed to several real-world projects under the guidance of experienced mentors. I worked on backend systems using **Java Spring Boot** and **.NET Core**, and created frontend interfaces with **Angular**. In the **Data Engineering** track, I built pipelines and workflows using **Apache Kafka**, **Airflow**, and **PySpark on Databricks**. I also gained hands-on experience with tools like **Docker**, **Git**, **GitHub**, and **GCP** for deployment and automation.

A highlight of the internship was the **LangGraph** project, where I explored AI-assisted frontend development using **Llama 3 Vision** to convert SRS and UI screenshots into Angular components. I also learned to test and validate APIs and UI flows using **Postman**, **Swagger**, and **Cypress**.

The internship fostered strong teamwork and communication, with daily interactions and reviews that helped me grow both technically and professionally. It gave me a deep understanding of how software products are built in real-world settings and prepared me to take on complex engineering challenges with confidence.

1.3 Overall Description of the work carried out by the team

The internship team was part of the **HU University Program**, a structured training initiative by **HashedIn by Deloitte** aimed at providing real-world software development experience. The program followed Agile methodologies and was organized into technical tracks such as **full-stack development**, **data engineering**, **DevOps**, and **AI-assisted development**. Each week functioned as a sprint with focused learning objectives, deliverables, and team collaboration.

The team worked on a variety of projects, starting with the development of a **social networking platform (SOCIO)**. The backend, built using **Java Spring Boot**, involved creating RESTful APIs, implementing **JWT authentication**, and managing data with **MySQL** using JPA/Hibernate. The frontend team used **Angular** to build responsive UIs, implement form validation, and secure route navigation. These components were integrated to form a fully functional web application.

Another team project involved a **blogging platform using .NET Core**, focusing on secure user authentication, blog and comment management, and role-based access control. The application followed clean architecture principles, with testing and documentation handled via **Postman** and **Swagger**.

Interns in the **Data Engineering** track focused on building **real-time and batch data pipelines** using **Apache Kafka**, **Apache Airflow**, and **PySpark on Databricks**. They developed ETL workflows, handled large datasets, and monitored data consistency across processing stages.

A notable collaborative effort was the **LangGraph project**, where the team explored **AI-assisted frontend generation** using **Llama 3 Vision**. The AI-generated Angular components were integrated into existing apps, debugged using **LangSmith**, and tested with **Cypress**, offering valuable insights into the role of AI in modern development.

Throughout the internship, the team used **GitHub** for version control and task tracking, **Docker** for containerization, and **Google Cloud Platform (GCP)** for deployment. Regular

code reviews and team discussions promoted active learning and a collaborative environment.

Overall, the work carried out by the team was diverse and hands-on, providing solid exposure to modern technologies and development workflows. It enhanced both technical proficiency and soft skills, preparing interns for real-world software engineering roles.

1.4 Overall Description of the work carried out by the team

As a **Trainee Engineer** in the **HU University Program** at HashedIn by Deloitte, I was involved in a structured, project-based internship designed to simulate real-world software development environments. The role gave me practical exposure to full-stack development, DevOps workflows, cloud deployment, data engineering, and AI-assisted UI generation.

My responsibilities included developing secure and scalable backend services using **Java Spring Boot** and **.NET Core**. I created RESTful APIs, managed data with **MySQL** and **Entity Framework**, and implemented **JWT-based authentication** with role-based access. I also worked on clean architecture principles, exception handling, and validation layers using tools like **FluentValidation**.

On the frontend, I built responsive interfaces using **Angular**. This included creating reusable components, integrating APIs, securing routes with guards, and enhancing the user experience with Angular Material, loaders, and toast notifications. I also contributed to API testing and documentation using **Postman** and **Swagger**.

As part of the **DevOps and deployment tasks**, I containerized applications using **Docker** and deployed them on **Google Cloud Platform**. I also used **GitHub Actions** for CI/CD automation, gaining hands-on experience in managing build and deployment pipelines.

A unique part of my internship involved working on the **LangGraph** project, where I used **Llama 3 Vision** to generate Angular components from UI mockups and requirement documents. I integrated and debugged the components using **LangSmith**, and validated UI flows through **Cypress** test cases.

In the final phase, I worked on **data engineering**, using **Apache Kafka**, **Airflow**, and **PySpark on Databricks** to build real-time pipelines and orchestrate ETL workflows. This involved working with DAGs, handling streaming data, and transforming datasets at scale. Throughout the internship, I collaborated in Agile sprints, participated in daily stand-ups, used **GitHub** for version control, and engaged in peer code reviews. These responsibilities helped me grow both technically and professionally, while giving me a strong foundation for real-world software engineering roles.

CHAPTER 2

TECHNOLOGY/TOOLS WORKED

2.1 List of Tools

• Docker

Description:

Used to containerize backend services and simplify deployments. Gained hands-on experience in creating Docker images, writing docker-compose files for multi-container setups, and deploying containers on GCP for scalable execution.

• Git and GitHub

Description:

Used extensively for version control and collaborative development. Followed GitFlow branching strategies, created pull requests, resolved merge conflicts, and tracked issues using GitHub Projects. Also configured GitHub Actions for CI/CD workflows.

• Java Spring Boot

Description:

Developed the backend for a full-fledged social network platform called SOCIO. Created RESTful APIs, implemented user authentication using JWT, added role-based access control, and handled exception management efficiently.

• Angular

Description:

Worked on the GroceryStoreMangement frontend using Angular. Created reusable components, implemented reactive forms with validation, integrated backend APIs, and used Angular Material for enhanced UI design. Route guards and role-based directives were used for access control.

• .NET Core (C#)

Description:

Built a blogging platform using ASP.NET Core. Structured the project using clean architecture, developed secure endpoints with JWT, applied FluentValidation for input models, and handled role-based access control for Admin, Blogger, and Subscriber roles.

• Data Engineering (Apache Kafka, Apache Airflow & PySpark on Databricks)

Description:

As part of the Data Engineering track, I worked extensively with tools used for building and orchestrating scalable data pipelines. Using **Apache Kafka**, I developed real-time data pipelines to simulate social media activity streams, gaining a solid understanding of producers, consumers, topics, partitions, and message flows in distributed systems. With **Apache Airflow**, I designed and scheduled automated ETL workflows by creating DAGs (Directed Acyclic Graphs), handling task dependencies, retries, and monitoring executions. On **Databricks**, I utilized **PySpark** to process structured datasets, applying transformations, aggregations, and data cleaning operations. I worked with both RDDs and DataFrames, and leveraged Spark SQL for querying large datasets efficiently. Together, these tools provided hands-on exposure to the full lifecycle of modern data engineering—ingestion, processing, orchestration, and analytics

• LangGraph + LangChian

Description:

Used LangGraph to automate frontend component generation based on uploaded SRS documents and UI mockups. Leveraged Llama 3 Vision to extract design requirements and generate structured Angular code. Debugged AI-generated outputs using LangSmith feedback

• Postman and Swagger

Description:

Tested RESTful APIs using Postman by simulating client requests. Swagger was used to auto-generate interactive documentation and validate endpoints during integration and testing.

CHAPTER 3

DESCRIPTION OF THE WORK CARRIED OUT

3.1 Introduction

This chapter provides an in-depth overview of the various projects, tools, and technologies I worked on during my internship at **HashedIn by Deloitte**, conducted through the **HU University Program**. The internship was structured as a bootcamp-style learning and development experience tailored to simulate real-world software engineering workflows. It emphasized industry best practices, hands-on coding, end-to-end project development, and collaborative team dynamics. Each week was carefully planned to introduce a new set of technologies and responsibilities, allowing me to build a diverse technical foundation while applying classroom knowledge to real-world scenarios.

The program offered focused exposure to critical domains such as **full-stack web development**, **DevOps**, **cloud deployment**, **data engineering**, **AI-assisted development**, and **enterprise-grade application security**. Through practical sprints and mentor-led guidance, I learned how to contribute meaningfully to cross-functional software teams, write scalable and maintainable code, and manage real-time deployments on cloud platforms.

As a **Trainee Engineer**, I was exposed to the complete **software development lifecycle (SDLC)**—starting from requirement analysis and planning, followed by application design, coding, integration, testing, deployment, and iterative feedback. The internship followed an Agile-based structure, with weekly objectives, daily stand-ups, task tracking on GitHub Projects, and regular peer reviews. This iterative process helped me become more organized, accountable, and collaborative.

The internship also offered a multidisciplinary experience. I was able to rotate across different technical stacks and domains—from backend API development with **Java Spring Boot** and **.NET Core**, to frontend application design using **Angular**, to large-scale data processing with **Apache Kafka**, **Airflow**, and **PySpark**. I also got the opportunity to work on **generative AI** development using **LangGraph**, integrating AI-generated UI components into a real project environment.

Throughout the internship, I was challenged to move beyond theory and apply concepts through practical coding. I gained confidence in using tools like **Docker**, **Git**, **GCP**, **Postman**, **Swagger**, **LangSmith**, and **Cypress**, which are widely used in modern development pipelines. These experiences significantly strengthened my technical, analytical, and problem-solving skills, while also giving me insight into industry expectations and workflows.

This chapter outlines the weekly breakdown of the internship, detailing the projects, technologies used, and the outcomes of each phase. It highlights the structured progression from core software development to advanced topics like AI automation and data engineering—providing a comprehensive view of my professional growth during the internship.

3.2 Write-up on the work

The internship began with foundational sessions in **DevOps**, **Agile development practices**, and **version control systems**. In the first week, I was introduced to two essential tools: **Docker** and **Git**. I learned how to containerize applications using Docker, making them platform-independent and easier to deploy. Using docker-compose, I configured multi-container environments and understood how services communicate over isolated networks. I then deployed these containers on **Google Cloud Platform (GCP)** using **Compute Engine**, learning to manage VM instances, firewall rules, and deployment configurations.

In parallel, I worked with **Git** and **GitHub**, gaining a deep understanding of **GitFlow branching strategies**, pull requests, merge conflicts, and version control best practices. I used **GitHub Projects** for task tracking and participated in code reviews, which gave me insight into collaborative development and team-based workflows. This initial training laid a solid foundation for the upcoming weeks, where I would apply these practices in real projects.

In the **second week**, I began working on the backend of **SOCIO**, a full-fledged social networking platform. The application was built using **Java Spring Boot**, and I was responsible for setting up the project structure, configuring databases, and building core backend modules. I created APIs for user registration, login, and profile management. I implemented **JWT (JSON Web Token)**-based authentication and integrated **Spring Security** to enforce access restrictions based on user roles. As the project progressed, I developed endpoints for post creation, updating, deletion, and post interactions such as likes and comments. I also implemented user relationship features like follow and unfollow, giving the platform a realistic social network feel.

By the end of the second week, I had gained hands-on experience with **entity relationships**, **ORM (Object Relational Mapping)** using **Hibernate**, and **database schema design** for entities like User, Post, Comment, and Group. I also configured global exception handling, response wrapping, and status codes for better API communication.

The **third week** marked the shift to frontend development using **Angular**. I began working on the **Grocery Store Management** frontend module, which involved building responsive layouts, dashboards, and user interfaces with **Angular Material**. I implemented reactive forms for login and registration, including validation logic and error handling. Integration with backend APIs was achieved using Angular's HttpClient, and I created services for authentication, token handling, and session management.

I applied **route guards** to restrict access based on authentication state and user roles. Additionally, I used Angular's directive-based rendering to conditionally display UI elements for Admins, Managers, and Users. I enhanced the user experience by integrating **toast notifications**, **loading indicators**, **modal dialogs**, and **route animations**. This week helped me understand the importance of user-centric design, API integration, and frontend state management in building scalable client-side applications.

The **fourth week** introduced me to the intersection of AI and software development through the **LangGraph project**. This project showcased how **generative AI**, particularly **Llama 3 Vision**, can analyze **SRS (Software Requirement Specification)** documents and UI mockups to generate frontend components. I uploaded SRS files and screenshots, and LangGraph generated modular Angular code based on inferred structure and expected behavior. I ensured that these components adhered to best practices like separation of concerns, reusability, and accessibility standards.

After generating the components, I integrated them with backend APIs and used **LangSmith** to debug and trace AI-generated workflows. I conducted **performance benchmarks**, and automated UI testing was done using **Cypress**, where I learned how to create end-to-end test cases to validate user flows, input validation, and route transitions. This project broadened my perspective on how AI can significantly accelerate frontend development and reduce repetitive manual work.

In the **fifth week**, I moved to a different stack and worked on a **Blogging Platform** using **.NET Core and C#**. I developed secure REST APIs for user registration, login, role assignment, and CRUD operations for blog posts and comments. The platform supported different user roles—Admin, Blogger, and Subscriber—with access control implemented using JWT-based authentication and custom middleware logic. I used **Entity Framework Core** for database interactions and **FluentValidation** for model validation.

I also configured **Swagger** for API documentation and tested endpoints using **Postman**. I implemented admin-level controls such as user banning, post moderation, and role reassignment. This project deepened my understanding of backend layering, API security, DTOs, and dependency injection in .NET applications.

The **final week** was focused on **Data Engineering**, where I gained exposure to modern data processing tools and platforms. I started with **Apache Kafka**, where I created topics and simulated real-time data streams, learning how producers and consumers interact in a distributed message-driven architecture. I then used **Apache Airflow** to build and schedule **ETL (Extract, Transform, Load)** pipelines. I designed DAGs (Directed Acyclic Graphs) to automate workflows and manage task dependencies, retries, and success/failure conditions.

On **Databricks**, I worked with **PySpark** to process structured datasets using both **RDDs** and **DataFrames**. I applied transformations, filtering, aggregation, and joins to clean and analyze large volumes of data. I also experimented with **Spark SQL** to query datasets and perform analytics at scale. This experience helped me understand how big data is handled in production systems, and how data engineering supports modern applications and business intelligence.

CHAPTER 4

PERSONAL ROLES AND OBSERVATION

During the course of my internship, I served as a **Trainee Engineer**, actively contributing to multiple full-stack and data-oriented projects. My responsibilities spanned across the development lifecycle—ranging from frontend design and backend logic to deployment and automation. I worked on building secure RESTful APIs, developing responsive user interfaces, integrating various services, and participating in team-based problem-solving sessions. By engaging in both backend and frontend tasks, I gained a holistic understanding of how software systems are architected, developed, and deployed in a real-world setting.

One of the most significant takeaways from this internship was the improvement in my **technical proficiency and problem-solving skills**. I became more confident in writing clean, maintainable code, particularly in **Java (Spring Boot)** and **Angular**, and also learned how to effectively debug issues related to logic, integration, and deployment. I applied **secure authentication practices**, using **JWT** and **Spring Security**, and implemented **role-based access control**, which are critical skills for developing production-ready applications.

Working with tools like **Postman**, **Swagger**, and **Cypress** helped me understand the importance of API testing and automation. I also became familiar with **CI/CD workflows** using **Docker** and **GCP**, gaining insight into how continuous integration and deployment streamline software releases and maintenance. The experience of orchestrating microservices and containerized apps enhanced my understanding of cloud infrastructure and DevOps principles.

Being part of Agile-based development cycles through **GitHub Projects** taught me how to manage tasks, collaborate on version control, and handle peer reviews. I learned how to effectively communicate with mentors and teammates, discuss issues openly, and incorporate feedback constructively. One of the most rewarding aspects of the internship was participating in **AI-assisted development** using **LangGraph**, where I saw how generative AI can dramatically accelerate frontend development and improve code quality through automated testing and intelligent suggestions.

In the **Data Engineering** track, I worked with **Apache Kafka**, **Airflow**, and **PySpark**, which expanded my skill set beyond application development. I learned how real-time and batch data pipelines are built, how ETL processes are orchestrated, and how massive datasets are cleaned, transformed, and analyzed efficiently. This experience gave me a strong foundation in the data lifecycle and how it integrates with application ecosystems.

I encountered several technical and collaborative challenges along the way—such as version conflicts, CORS issues, and database migrations—but overcame them through structured debugging, active mentorship, and perseverance. These situations helped me become more resilient and analytical, qualities that are essential for any software engineer.

Overall, the internship was a transformative experience that contributed immensely to my **personal and professional growth**. It taught me how to think critically, work as part of a team, write production-level code, and approach problems methodically. The exposure to diverse technologies, combined with hands-on mentorship and real-world projects, prepared me to step confidently into the world of software engineering.

CHAPTER 5

CONCLUSION

The internship at **HashedIn by Deloitte**, through the **HU University Program**, was a transformative experience that bridged the gap between classroom learning and real-world software engineering. Over six weeks, I gained practical exposure to the complete software development lifecycle—covering planning, development, testing, deployment, and optimization—using modern tools and industry best practices.

I began by learning **DevOps fundamentals**, containerization with **Docker**, and deploying applications using **Google Cloud Platform**. The **SOCIO** project helped solidify my skills in full-stack development using **Spring Boot** for the backend and **Angular** for the frontend, implementing features like JWT authentication, role-based access, and user interactions.

Working on the **LangGraph** project introduced me to **AI-assisted development** using **Llama 3 Vision**, where I explored how frontend components can be auto-generated from design inputs. This provided insights into the evolving role of generative AI in software development.

In the later phase, I focused on **Data Engineering**, working with **Apache Kafka**, **Airflow**, and **PySpark on Databricks** to build real-time pipelines, orchestrate workflows, and process large datasets. This gave me a solid foundation in data-driven development alongside application engineering.

Throughout the internship, I improved my coding, debugging, and documentation skills, learned to work in Agile teams, and adapted to rapidly changing tech environments. More importantly, I realized the importance of curiosity, consistency, and collaboration in becoming a successful developer.

This internship has prepared me to tackle real-world engineering challenges with confidence and has inspired me to explore advanced areas like AI and data engineering further in my career