

Internship at
Advanced Software Development (ASD):
Software Engineer Trainee

by

Sabiha Tahseen
23341025

An internship report submitted to the Department of Computer Science and
Engineering
in partial fulfillment of the requirements for the degree of
Computer Science and Engineering

Department of Computer Science and Engineering
School of Data & Sciences
Brac University
January 2024

© 2024. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The internship report submitted is my original work while completing my degree at Brac University.
2. The report does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The report does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

Sabiha Tahseen

Sabiha Tahseen
23341025

Approval

The internship report titled “Internship at Advanced Software Development (ASD): Software Engineer Trainee” submitted by

1. Sabiha Tahseen (23341025)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 9, 2024.

Examining Committee:

Supervisor’s Full Name and Signature:
(Member)

Annjiat Alim Rasel
Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Industrial Supervisor’s Full Name and Signature:
(Member)



Tanvir Mahatab
Senior Software Engineer
Advanced Software Development

Internship Coordinator’s Full Name and Signature:

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

This report has been authored by me, and it is presented with the assurance that it is entirely free of plagiarism. While I have gathered some information from the company's website, these sections have been appropriately cited and referenced. I have adhered to ethical writing practices throughout the report to ensure academic integrity. I have maintained the confidentiality of sensitive information and respected the intellectual property of the organization.

Abstract

During my six-month internship at ASD, I embarked on an exciting journey of learning and hands-on experience. In the initial four months, I dived into the world of Restful Web API and managing Microsoft SQL Server databases. This involved creating APIs, working with database tables, and utilizing Swagger for effective API testing. A big part of my internship was about getting really good at using Postman, this cool tool for testing and working on APIs together with a team. As time passed, I also started learning about microservices. I faced challenges as the company used many tools and technologies. Despite challenges, I saw them as learning opportunities. I tried to learn all of the tools and technologies. This report summarizes what I have learned and the things I have been working on at ASD.

Keywords: Restful Web API ; Microsoft SQL Server ; Swagger ; Microservices ; Postman

Dedication

This report stands as a testament to the countless hours of effort and dedication poured into my internship at ASDBD. I also dedicate it to the importance of APIs and their role in modern technology.

Acknowledgement

To begin, I express my gratitude to the Almighty Allah for guiding me through six months of my internship without significant disruptions.

Next, I extend my appreciation to my supervisor, Annajiat Alim Rasel sir, for his generous support and valuable guidance throughout my work. His assistance was always available whenever I required it.

Furthermore, I am grateful to my industrial supervisor, Tanvir Mahatab, who provided guidance and support throughout my entire internship duration.

Last but not least, I want to express my heartfelt gratitude to my parents, whose unwavering support and prayers have brought me to the brink of graduation.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	ii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
Nomenclature	xii
1 Introduction	1
1.1 About Internship	1
1.2 Internship Objective	1
1.3 Problem Statement	2
1.4 Methodology	2
1.4.1 Primary Data Sources	2
1.4.2 Secondary Data Sources	2
2 Company Profile	3
2.1 Overview	3
2.2 Vision	3
2.3 Achievements	4
2.4 Services Offered	4
2.5 Industry That ASD Cover	5
2.6 ASD Products	5
2.6.1 Banking Application	5
2.6.2 Others Solution	10
2.6.3 IT Enabled Service	13
2.7 Scalability	13
2.8 Authentication and Authorization	13
2.9 Data Encryption and Key Management	13
2.10 Logging and Monitoring	14

2.11	Technology Stack	14
2.12	ASD Clients	14
3	Workplace Details	15
3.1	Management and Corporate Structure	15
3.2	Development Environment and Team	16
3.3	Software Development Process	17
3.4	Work Environment	17
3.5	Communication and Meetings	18
4	Training Phase	19
4.1	Overview	19
4.2	Getting familiar with ASDBD's Application Architecture	20
4.3	Environment Setup for Web API Development	21
4.4	.NET SDK	21
4.5	Package Manager	21
4.6	ASP.NET Core	21
4.7	Setup Visual Studio	22
4.8	Building a Restful Web API	25
4.9	Swagger API	28
4.10	Postman	30
4.11	Postman vs. Swagger: Uncovered	33
4.12	Microsoft SQL Server	33
4.13	Tutorials and Assistance	38
4.14	Learning Path	39
5	Conclusion	40
5.1	Summary	40
5.2	Future Work	40
	Bibliography	41

List of Figures

2.1	ASD Major Clients	14
3.1	ASD Organogram	15
3.2	ASD Development Team	16
4.1	ASD Microservice Architecture	20
4.2	Microservices with Clean Architecture	20
4.3	Visual Studio 2022 User Interface	22
4.4	The Administration Folder	23
4.5	The Controllers Folder	24
4.6	‘BusinessProfileController’ Class	25
4.7	Short Code Generating API	26
4.8	API For Short Code Existence	26
4.9	User Sign Up API	27
4.10	Email Verification API	27
4.11	Nationality List Collection API	28
4.12	Swagger API Page	29
4.13	BusinessProfile On Swagger	29
4.14	Postman Testing For Short Code Generating API	30
4.15	Postman Validation Of API For Short Code Existence	31
4.16	Postman Testing - User Sign Up API	31
4.17	Postman Testing - Email Verification API	32
4.18	Postman Testing - API for Nationality List Collection	32
4.19	Microsoft SQL Server Management Studio	33
4.20	Databases Folder	34
4.21	InfiniumSuit_Administartion Folder	35
4.22	Tables Folder	36
4.23	system.BusinessProfiles Folder	37
4.24	Selecting Top 1000 Rows	38
4.25	My Learning Path	39

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ABP Auto Bills Pay

AES Advanced Encryption Standard

API Application Programming Interface

ASD Advanced Software Development

ATM Automated Teller Machine

BI Business Intelligence

BTRC Bangladesh Telecommunication Regulatory Commission

BVS Biometric Verification System Platform

CDR Call Detail Record

CIB Credit Information Bureau

CPS Cheque Processing Suite

CQRS Command and Query Responsibility Segregation

CRM Customer Relationship Management

CSRA Corporate SIM Registration & Approval Management System

DESCO Dhaka Electric Supply Company Limited

DPDC Dhaka Power Distribution Company Limited

e – KYC Electronic Know Your Client

eBBS electronic basic banking system

EOS Enterprise Office Solution

ERP Enterprise Resource Planning

ESAF External Subsystem Attach Facility

FCY Foreign Currency

goAML Anti-Money-Laundering System

HR Human Resources

HRIS Human Resources Information System

IBM International Business Machines

IDTP Interoperable Digital Transaction Platform

IMEI International Mobile Equipment Identity

IMMS Interface, Management and Monitoring System

IMSI International Mobile Subscriber Identity

LINQ Language Integrated Query

MFS Mobile Financial Services

MIS Management Information System

NASCO North American Students of Cooperation

NBR National Board of Revenue

NITA Non-resident investor's Taka Account

npmS Node Package Manager

NTMC National Telecommunication Monitoring Cell

NTMC National Telecommunication Monitoring Centre

OOP Object-Oriented Programming

PSOs Program Specific Outcomes

PSPs Payment service providers

PuTTY terminal emulator application

SABINCO Saudi Bangladesh Industrial and Agricultural Investment Company Limited

SCB Standard Chartered Bank

SHA – 256 Secure Hash Algorithm 256-bit

SLA Service-Level Agreement

SND Special Notice Deposit

SSL Secure Sockets Layer

SSMS SQL Server Management Studio

TITAS Taipei Innovative Textile Application Show

TPS Transaction Processing Systems

TRANSIT Technology Review Assistance Notification Standards Integration and Testing

TXN Transaction

UDR Unified Data Repository

WASA Water Supply and Sewerage Authority

Chapter 1

Introduction

1.1 About Internship

This report offers an overview of my internship experience at ASDBD (Advanced Software Development), a prominent software company specializing in banking software, banking apps, ERP solutions, and more. The internship provided a valuable opportunity for me to apply my knowledge in a practical work environment. Throughout this report, I will discuss the purpose of the internship, highlight the significance of bridging academia and the professional world, and outline the objectives and structure of the report. Working at ASDBD exposed me to industry professionals, allowed me to acquire new skills, and witness the implementation of cutting-edge technologies. This report serves as a comprehensive account of my internship, demonstrating the practical application of theoretical concepts and its relevance to my career development. Internships are increasingly popular among students, with some choosing them over writing a thesis due to gaining practical industrial knowledge, which may be advantageous for both higher studies and future jobs. At BRAC University, students can choose between completing an internship or a project or a thesis. The university actively supports and encourages students to acquire valuable workplace experience. I chose the internship path, fully aware of its significance in gaining practical skills, networking opportunities, and real-world exposure. This opportunity allowed me to select a specialization relevant to my major and engage in a minimum 6-month internship with a reputable company.

1.2 Internship Objective

The objective of my internship is to develop a thorough grasp of the company's software development process and the functionality of its software application. As a newcomer to the field, my main responsibility is to closely observe and become well-versed in the various components and features of the software as a preparation for fully engaging in day to day activities of a software firm. I was asked by the software architect Tanvir Mahatab to comprehend how their software application works. This report signifies the completion of my six-month internship at the company, during which I had the opportunity to explore and learn from experienced professionals. Recognizing that expectations are modest for freshers in software companies, I am enthusiastic about diligently undertaking assigned tasks and demonstrating a strong

commitment to learning and personal growth. Through this internship, I aim to expand my technical knowledge and improve my problem-solving abilities.

1.3 Problem Statement

In today's software world, using Restful Web APIs are crucial. These APIs help different software parts talk to each other. But, linking these APIs with Microsoft SQL Server, where data is stored and managed, can be tricky [6]. Making sure the data stays safe while the API works well is one challenge. Plus, using Swagger to test and explain the API adds more to the mix. This report tries to demonstrated the process how I handled these things and learned about APIs, SQL Server, and Swagger along the way [7].

1.4 Methodology

This report primarily focuses on my work experience and the lessons I learned during my internship at ASDBD Ltd. The majority of the information presented is based on my personal observations and experiences. Additionally, I have also gathered some information from specific websites. The data collected is primarily sourced from the following:

1.4.1 Primary Data Sources

Primary data sources for the report include:

1. Personal practical experience gained from working.
2. Invaluable insights and knowledge acquired through interactions with colleagues and team leaders,
3. Insights obtained from attending seminars and participating in team meetings.

1.4.2 Secondary Data Sources

Secondary data sources for the report include:

1. ASDBD Official Website
2. Online documentations, books, etc. related to relevant tools and technologies.

Chapter 2

Company Profile

2.1 Overview

ASDBD (Advanced Software Development) is a leading software company based in Dhaka, Bangladesh. With a strong focus on delivering exceptional software solutions, the company specializes in providing top-notch banking software, banking apps, ERP solutions, and more. ASDBD's team is driven by a passion for technology and a profound understanding of business needs. They strive to exceed client expectations by ensuring that their software engagements align with both the business and technology requirements. Their benefits and opportunities are flexible workspace, knowledge sharing, competitive remuneration, meeting room, growth opportunity, inspiring culture, lunch, snacks and coffee, game zone, fitness center. They have around 250 happy customers, 50 experienced developers, 360 degree technical coverage, 19 years of experience.

2.2 Vision

ASDBD aspires to achieve excellence in software development and aims to be among the top ten fastest-growing companies in Asia by 2030. This company is committed to delivering high-quality software solutions that meet both business and technology expectations while adhering to timelines and budgets. With a strong passion for technology and deep industry expertise, ASDBD strives to establish itself as a leader in the field. The company's focus on innovation, customer satisfaction, and embracing emerging technologies sets it apart and drives its growth. ASDBD aims to contribute to the overall growth and success of the software industry in Asia while maintaining strong client relationships and a forward-thinking mindset.

2.3 Achievements

The company has achieved remarkable milestones in its journey:

- 1. Client Success Stories:** Our company's creative software solutions have supported many clients in reaching their business objectives. This has led to better operational efficiency, improved customer experiences, and increased profitability.
- 2. Industry Recognition:** Our company has received awards and acknowledgment for its excellent work in software development.
- 3. Talent Development:** ASD support local talent. Our company supports the growth of tech skills in Dhaka by training and developing local talents.

2.4 Services Offered

- 1. Web Design:** In ASD, we make awesome websites that fit our client's business perfectly. Our skilled team uses the latest tech to create sites that look good and are easy to use.
- 2. Software Development:** Our aim is to give clients easy-to-use computer programs that work great for their businesses and we are known for being reliable, flexible, high-performing, and for giving excellent customer service.
- 3. Mobile Application Development:** Our smart team uses the newest tools to make top-notch mobile apps that work on iPhones, Androids, and other devices.
- 4. Web Hosting:** ASDBD offers reliable and high-quality web hosting services for businesses of all sizes. Our hosting packages are designed to meet your specific needs, whether you're starting out or need an upgrade.
- 5. E-Media Solutions:** Establishing a strong online presence is crucial in today's digital age. ASDBD offers comprehensive E-Media services to help businesses effectively engage their target audience and build a robust online presence.
- 6. E-commerce Solutions:** ASDBD provides comprehensive E-commerce services to help businesses thrive in the online marketplace. Our tailored solutions ensure a seamless shopping experience for your customers while streamlining your business processes.

2.5 Industry That ASD Cover

1. Government Sector
2. NGOs
3. News Paper Houses
4. Banking and MFS
5. Non-Government Organization
6. Medical and Pharmaceuticals
7. FMCGs
8. Telecommunications
9. E-commerce
10. Real-Estate
11. University
12. Hotel and Tourism

2.6 ASD Products

Advanced Software Development has developed a range of innovative software products catering to various industries. Some notable products include:

2.6.1 Banking Application

1. CPS - Cheque Processing Suite

Cheque Processing Suite (CPS) is the complete solution for managing Clearing Cheques, Positive Pay and House Transfer Cheques. With built-in OCR, automated technical verification, automatic queuing and positive pay automation, ASD CPS is the state-of-the-art solution that saves time and cost of Cheque Clearing and increases productivity and customer satisfaction.

- Platform: Microsoft .Net Framework 4.7, .Net Core 3.1, C sharp, Angular 8, Microsoft SQL Server 2016/2019
- Representatives Clients: Standard Chartered Bank

2. Auto Bills Pay

Auto Bills Pay (ABP) is an end-to-end solution for paying utility bill payments (DPDC, DESCO, TITAS, NASCO, WASA etc.), Insurance Premium, Education institute's monthly fees, ISP bills, Club subscription, Hospital executive subscriptions, Mobile Post-Paid Mobile Bill recurring Payment will be done through AUTO-DEBIT Instruction from Account or Credit card without waiting in long queues. You can simply authorize paying all these bills directly from your Account or Credit Card every month through Auto Bills Pay.

- Platform: ASP.NET, SQL Server DB.
- Representatives Clients: Standard Chartered Bank, Brac Bank Limited

3. e-KYC Application

Our e-KYC Application provides a seamless and secure method for identity verification and onboarding processes. It connects with trusted systems to capture customer data accurately and efficiently.

- Platform: ASP.NET, SQL Server DB.
- Representatives Clients: Standard Chartered Bank, Habib Bank Limited.

4. IDTP IMMS

IDTP (Interoperable Digital Transaction Platform) IMMS – IDTP Interface, Management and Monitoring System platform is a processing and settlement system for electronic payments, where the main participants are Bank, MFS and PSPs. IDTP will also connect the utility service providers, other Govt. entities that collect fees and charges from individuals and corporates and also have the provision to connect the business entities such as e-commerce merchants and the PSOs therefore, it will act like a Master Payment Gateway.

- Platform: Microsoft .Net Framework 4.7, .Net Core 3.1, C #, Angular 8, Microsoft SQL Server 2016/2019
- Representatives Clients: Standard Chartered Bank.

5. Sanchayapatra Patra Management System

Sanchayapatra is a Central bank product. All banks in Bangladesh are bound to sell this product. There are many kinds of Sanchayapatra product. This system manages all criteria of central bank & perching banks rules.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Hong Kong and Shanghai Banking Corporation Ltd.

6. Export Import Management System

The Export Import Management System facilitates the issuance of letters of credit and effectively manages import-export transactions. It ensures a secure and standardized process between buyers and sellers. The system is web-based and supports multiple users, with a customizable entry module. It generates management information system (MIS) and regulatory reports.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Hong Kong and Shanghai Banking Corporation Ltd.

7. Service Request Tracking System

The major objective of the solution is to automate the tracking of service requests. The system will track the requests from issuance to delivery. During the process, it will check the SLA and trigger some data to get custom MIS.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Hong Kong and Shanghai Banking Corporation Ltd. East-ern Bank Ltd.

8. Loan Collection & Recovery System

Our comprehensive debt receivables management system streamlines collection operations for loans, credit cards, and overdrafts, reducing delinquencies through proactive actions and supervision.

- Platform: ASP.NET Desktop-based Application, SQL Server DB.
- Representatives Clients: Standard Chartered Bank, Premier Bank Ltd. SABINCO.

9. Tracker for Loans In Transit

TRANSIT will be a workflow-based loan origination and processing system, which will help Premier Bank to differentiate from the competition. It provides a comprehensive view of the customer and supports various financial products. The system enables multi-level approvals, integrates with Document Management & Imaging, and offers customizable MIS reports.

- Platform: ASP.NET Desktop-Based Application, SQL Server DB.
- Representatives Clients: Standard Chartered Bank, Premier Bank Ltd. SABINCO.

10. Complaint Management System

This solution that automates customer complaint management. Here, system Process flow will be parameterized so that Process flow can be changed by Administrator, incorporation of Dynamic fields in Managing of customer complaint and customers' complaint status with escalation will be viewable at any point of accessing the system

- Platform: ASP.NET Desktop-Based Application, SQL Server DB
- Representatives Clients: Standard Chartered Bank, Premier Bank Ltd, Eastern Bank Ltd.

11. Cash Transaction Report System [goAML]

Our system does the submission of cash transaction reports to the Central Bank for the Cash Deposits or cash Withdrawals of BDT Tk. 1,000,000 or above in any account in a day.

- Platform: Java Web-based Solution/J2EE, SQL Server DB
- Representatives Clients: Standard Chartered Bank.

12. Card Production System

The system pulls out the necessary info from the core banking system to manage the card production.

- Platform: Java Web-based Solution/J2EE, SQL Server DB
- Representatives Clients: Standard Chartered Bank.

13. Certificate Automation System

This system generates various certificates for BANK customers. Like balance, mortgage, loan, solvency, bond etc.

- Platform: Java Web-based Solution/J2EE, SQL Server DB
- Representatives Clients: Standard Chartered Bank.

14. Check Printing Software

The check printing software is a solution that simplifies and automates the process of printing checks for companies and small businesses. It offers features such as multiple currency support, customizable templates, and enhanced security, while also generating reports for check writing history and tracking.

- Platform: ASP.NET Desktop-Based Application, SQL Server DB.
- Representatives Clients: Standard Chartered Bank, Hong Kong and Shanghai Banking Corporation Ltd.

15. CMO Suite—Process Automation System

This common platform software can manage all activities. User roll-based system.

- Platform: Java Web-based Solution/J2EE, SQL Server DB
- Representatives Clients: Standard Chartered Bank.

16. Bills Pay Collection System

This system is a dual control system for automated deposits and bills payment through ATM. Machine-generated transaction logs will be imported to the system and reconciled with physical cash and cheque.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

17. Electronic Journal & ATM Reconciliation

This system is a shadow for all the Bank ATMs. All the transactions and events will be logged and monitored in the system for a faster query, reports and service without being present on the spot.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

18. Central Bank Exchange Position Reporting System

The Central Bank has to be reported from every bank for their every foreign currency exchange record. Our system helps to send this report by retrieving data from the core banking system.

- Platform: ASP.NET, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

19. Charge Realization Management System

The system calculates the required charges to the bank customer & makes a transaction file of the core banking system.

- Platform: Java Web-based Solution/J2EE, SQL Server DB
- Representatives Clients: Standard Chartered Bank.

20. NITA Automation System

Non-resident investor's Taka account reporting tools for the central bank. The major objective of the solution is to automate the process of customer stock handling and generate a formatted return report for submission to Bangladesh Bank. The system will be able to take transaction information in Text or in Excel as input, and will also provide the necessary input screen for data input. With the defined formatting, the system will be able to generate report for certain tenure.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Hong Kong and Shanghai Banking Corporation Ltd.

21. Positive Pay Management System

Reporting tools for central bank Positive Pay instruction.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

22. Statement Automation and CIB Database

Automatic statement generation & client communication system.

- Platform: ASP.NET Desktop-Based Application, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

23. Excise Duty Management System

The Excise Duty Management System efficiently manages the deduction of excise duty from customer accounts based on NBR instructions. It retrieves account data from the IBM DB2 UDR database and generates block upload files for excise duty. With dual control and multi-user functionality, it ensures secure and reliable operations. The system also allows the import of FCY account balances and information.

- Platform: Java Web-based Solution/J2EE, SQL Server DB
- Representatives Clients: Standard Chartered Bank.

24. Special Notice Deposit (SND) Interest Forfeit

SND is a web-based multi-user system that will be accessed from all branches and SCB offices and it automatically calculates interests for forfeit and generates monthly/periodic reports

- Platform: Java Web-based Solution/J2EE, SQL Server DB
- Representatives Clients: Standard Chartered Bank.

25. Priority Reward Engine

This system calculates reward collection & generates a core banking system unlovable file.

Functions:

- The System will be a web – based and multi-user system
- Datacenter user will upload the daily TXN file and month-end eBBS and ELending file
- Dynamic and Customizable scoring system
- Parameterized reward engine
- Card400 bonus import file generation following the Card400 specification
- Multi-layer access control:

Administrator – Creates Users, Sets up Parameter, Creates Scoring Logics
 Super Administrator – Approves admin Activities
 Maker – Process data
 Checker – Approves reward data
 Datacenter User – Upload TXN, BDDDB2 and ELending files
 • Customized MIS reports

26. SBS2 Reporting System

This system is a reporting tool for the central bank. As per central bank requirement system, prepare the sector code wise reporting.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

27. Smart Search Engine for Customer Data Inquiry

The system searches the duplicate data from central database & shows an accurate or nearest result for protecting fraud case.

- Platform: ASP.NET Desktop-Based Application, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

28. ATM Monitoring System SMS/Email Based

The status of all SCB ATMs are monitored by “Sparrow” kit, which logs all the status changes. From the desktop, support engineers can monitor the ATMs with the telnet application, ANZIOLITE or PUTTY. But it’s not viable to look at the monitor all the time and report to the person assigned to a particular ATM. SMS/Email-Based ATM Monitoring System does this job. This system sends an instant SMS/Email describing the ATM status to the concerned person.

- Platform: Java Web-based Solution/J2EE, SQL Server DB.
- Representatives Clients: Standard Chartered Bank.

2.6.2 Others Solution

1. Enterprise Office Solution-EOS

Key Features:

- Multiple User Access and permission
- Accounts, HRM, CRM, Sales, Purchase, and Inventory modules in a single software.
- Easy Customer Management, Product Management, and Supplier Management
- Live Monitoring of Accounts, HRM, CRM, Sales, Purchase, and Inventory
- Income Statement and Profit & Loss Statement Report
- Platform: Microsoft .Net Framework 4.7, .Net Core 3.1, C#, Angular 8, Microsoft SQL Server 2016/2019
- Representatives Clients: Sikder Group Limited, The Bagha Club, Standard Chartered Bank, Teletalk Bangladesh, HSBC Bank, Bangladesh Police, Confidence Group, Rohim Afrooz, SABINCO, Unique Group

2. Human Resource & Payroll Management

The Human Resource Management System streamlines various HR tasks, including employee communication, recruitment, training, motivation, tracking, assessment,

and rewards. The software features a database for inputting employee information, encompassing details like joining date, personal profile, education, and experience. This data is integrated into the Human Resource Information System (HRIS), enabling reporting and analysis. The HRIS also records employee department, location, designation, salary for payroll, and reporting hierarchy. ASD boasts a robust HRIS for creating and managing employee records.

- Platform: Microsoft .Net Framework 4.7, .Net Core 3.1, C#, Angular 8, Microsoft SQL Server 2016/2019

- Representatives Clients: Teletalk Bangladesh Limited, Asian Paints, Ezzy Automations Limited, IPDC Finance, ABC Radio, UNIQUE Group, dnet, Galaxy Bangladesh, Prothom Alo

3. Document Management System

Document Management System is a digitized archival system. The main function of the system includes scanning of paper-based documents of customer accounts along with static data, storing the image of documents in encrypted format, retrieving the stored images for verification / update together with user security management. The system has a very simple and user-friendly interfaces with faster accessibility. In every step, SCB defined security matrix has been followed.

- Platform: ASP.NET Desktop-Based Application, SQL Server DB

- Representatives Clients: Standard Chartered Bank, Premier Bank Ltd.

4. Club Management System

Features:

- Automates processes
- Enhanced data security
- Integration with other business systems
- Improved customer experience
- Drives revenues
- Platform: ASP.NET Desktop-Based Application, SQL Server DB
- Representatives Clients: Kingsuk

5. Sales Inventory System

Features:

- Web based and online multi-user system
- Barcode enabled sales and inventory
- Centrally controlled stock
- Stock movement from and to head office to branches and branches to branches
- Daily, Periodic, Monthly sales ledger
- Real-time stock status
- Sales with/without fixed/percentage discount
- Online shops like Amazon and others
- Inquiry books and Place orders online
- Online request for new books
- New books information and upcoming books
- Custom output for AccPac integration
- Multi-level user administration
- Secured database management and optimization

- Platform: Microsoft .Net Framework 4.7, .Net Core 3.1, C#, Angular 8, Microsoft SQL Server 2016/2019
- Representatives Clients: The Bagha Club

6. Biometric Verification System Platform – BVS

Features

- New SIM Registration (Prepaid)
- Verification of Corporate Users
- Re-registration
- SIM Change
- Transfer Ownership
- Deregistration
- New Postpaid Activation
- Campaign SIM Activation
- Representatives Clients: Teletalk Bangladesh Ltd

7. National Telecommunication Monitoring Cell – NTMC API Development

The system exposed a Web service through which NTMC will be able to get the data. This proposed system is designed by considering strong support on TPS, integration with several data retrieval sources, such as CRM, ESAF, IN, BI, Billing System, CDR & ERP etc. It is built by following open specifications like Java Enterprise Edition and some open technology will run on windows/Linux platform. The service tier will run on a Java EE standard web server connected with a database. Each user of the system will enjoy one or more roles. Based on the definition of the role, one user will be able to enjoy services from the system. The system exposed some APIs to NTMC as per requirement such as IMEI, IMSI, Radio Location, Last Call Location, CDR, SMS Content Query automation.

- Representatives Clients Teletalk Bangladesh Ltd. Robi Axiata Limited

8. Corporate SIM Registration & Approval Management System – CSRA

Features:

- Teletalk Interface for Corporate SIM Registration
- Initiation of Corporate SIM Registration
- Approval of Corporate SIM Registration
- Company Interface for verification and approval
- Company Interface for amendment
- Company Interface for Inquiry
- BTRC Interface for Approval
- BTRC Interface for Inquiry
- BVS Integration
- Integration with SIM Registration Mobile App
- Email and SMS Notification
- Reporting and Dashboard
- Representatives Clients: Teletalk Bangladesh Ltd.

2.6.3 IT Enabled Service

1. Digital Archive Service

We have developed a dynamic and most effective archival system for newspaper houses to digitalize the hard copy of newspapers. We have developed specialization in creating digital archival of files from original documents and printed version media through special methods and skills. Moreover, we always use professional grade equipment for high-quality image/document scanning. We store printed newspapers digitally and preserve the news in different categories by inputting News Title, Sub-Title, Page Number and Author Name as well. Therefore, our user-friendly system can find any news clippings/full-page newspaper within a moment from a strong and secured database. At the same time, we analyze the interest of your business and try to match the archival value of your materials and preserve it accordingly.

•Representatives Clients: Press Institute Bangladesh, Bangla Academy, Prothom Alo, Kaler Kantha, Banik Barta, Manabzamin, MRDI, Ajker kagaj.

2. Ad Monitoring Service

We track all kinds of Advertisements, Campaigns, Notice, Tender, e-Tender, Classified info etc. to disclose the scenario of advertisements aired by every individual advertiser, their competitors as well as the whole industry.

•Representatives Clients: The Daily Star, Asian Age

2.7 Scalability

Most of the mentioned applications are built on Microservice Architecture. Throughout our solution, the below features have been implemented for scalability and performance.

1. Cross-platform and container support
2. Asynchronous through async/await
3. Dependency Injection
4. SignalR and MQ
5. Cross-Site Request Forgery Protection
6. Innovative Action Filters
7. Extensible Output Caching Load Balancing Feature Ready to Expand Horizontally for Handling Large TPS and Performance Improvements.

2.8 Authentication and Authorization

The application has a built-in authentication service with Identity Server along with Microsoft Active Directory authentication. Application requires an SSL certificate for encryption and validation.

2.9 Data Encryption and Key Management

The applications use SHA-256 where applicable and AES-256 bit encryption for storing secured configuration and keys. The application is highly configurable and compliant with data security policies.

2.10 Logging and Monitoring

The applications have full-featured log management with standard technology implementation. All the activities and transactions are logged in the system properly, which is also viewed/monitored from the front-end with proper access right. Rich graphical dashboard to monitor system performance, components status and logs along with all transaction related widgets.

2.11 Technology Stack

ASD use different tools for various projects. Here are a few of them:

- Node.js 12.18.3 LTS
- Angular 8
- ASP.NET Core 3.1
- Microsoft SQL Server 2019
- Oracle 11g
- NGINX/NGINX Plus
- Rabbit MQ
- IMMS APIGW

2.12 ASD Clients



Figure 2.1: ASD Major Clients

Chapter 3

Workplace Details

3.1 Management and Corporate Structure

ASD has an excellent management and corporate structure, making it a well-organized company. The Figure 3.1 below shows evidence of that.

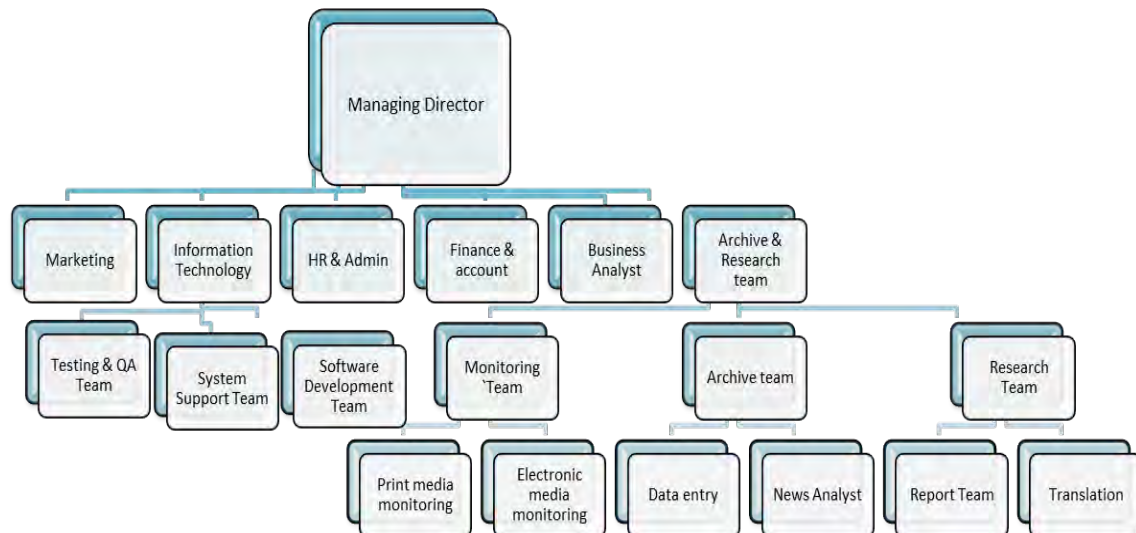


Figure 3.1: ASD Organogram

3.2 Development Environment and Team

How many people work in the development team, what technologies are used, are shown in Figure 3.2.

Development Environment				
Development Tools/Languages			Persons	
C#			7	
ASP.NET			4	
VB.NET			3	
Java			10	
PHP			4	
C/C++			2	
Database			Persons	
Oracle			5	
MS SQL Server			8	
My SQL			6	
DB2			3	
Role wise No. of Personnel				
Roles			Persons	
Project Manager			4	
Technical Lead			4	
System Analyst			3	
Architect			2	
Designer			2	
Developer			20	
QA Engineer			2	
GUI Developer			2	
DB Expert			2	
System Support Engineer			3	
Technology Skills				
Environment	Languages	Application Servers	Databases	Case Tools
Visual Studio, NetBeans, Eclipse, PhpEdit	C#, ASP.NET, VB.NET, Java, C/C++, PHP	IIS, Apache Tomcat, JBoss, Sun Application Server	MS SQL, My SQL, DB2, Oracle	Rational Rose, Case Studio

Figure 3.2: ASD Development Team

3.3 Software Development Process

ASDBD adopts Agile methodologies for its software development process. Agile emphasizes iterative and incremental development, allowing for flexibility and adaptability throughout the project lifecycle. Furthermore, some key points are:

1) Project Management: Project Management at ASDBD involves overseeing and coordinating software development projects [2]. The Project Manager acts as the main point of contact, ensuring effective communication and delivering high-quality solutions [3]. They work closely with the Product Owner to align development with project requirements.

2) Requirement Identification: ASDBD focuses on identifying and understanding the requirements and needs of their business team. This involves evaluating requests, brainstorming ideas, and documenting the identified requirements [1]. Various teams at ASDBD collaborate to provide solutions that align with the goals and expectations of the business team.

3) Context and Process Capturing: ASDBD initiates development by carefully researching various factors. A dedicated product manager analyzes the business context, resource needs, audience, competition, and project objectives [1]. This data helps create a development framework that guides the software architects in achieving a successful end product aligned with the business team's needs.

4) Eventual Releases: ASDBD employs Agile methodologies, particularly Scrum, for their development process. Emphasizing flexibility and collaboration, they work in short, iterative sprints. The Product Owner defines and prioritizes requirements, while the Scrum Master facilitates progress [2]. Daily stand-up meetings ensure alignment, and sprint reviews gather stakeholder feedback. This Agile approach promotes adaptability and continuous improvement in ASDBD's development process.

3.4 Work Environment

Working at ASDBD has been an engaging and enriching experience for me. I have had the opportunity to meet and connect with new colleagues, some of whom have become close friends in a short period of time. The work environment is supportive, with everyone being extremely helpful and open to ideas. From the moment I joined, I was treated as a valued member of the team, rather than just an intern, which facilitated a smooth transition into the company's work culture. While most employees typically perform onsite jobs, the eMedia department, consisting of around 10 to 15 employees, primarily works from home.

3.5 Communication and Meetings

At ASDBD, effective communication and regular meetings are key to our software project success. Our project manager holds a weekly meeting every Thursday, which serves as a platform to review the overall project progress, address any issues or concerns, and align our efforts for the upcoming week. We have various Skype chat groups dedicated to different departments, including development, HR, media, and marketing teams. These department-specific groups allow for focused discussions and collaboration within each team. Additionally, we have a general group that includes all departments, promoting cross-functional communication and fostering a sense of unity within the organization.

Chapter 4

Training Phase

4.1 Overview

During my six months at ASD, I primarily focused on learning the technologies used within the organization and examining project codes, providing a hands-on experience to understand both theoretical concepts and practical implementations. They use .Net Core version 3.1, Visual Studio 2022. According to company policy, they don't provide access to me in their GitHub account. They have one internal server and five cloud servers.

Although I already knew some technologies used there, including HTML, CSS, C#, Visual Studio, Git. There were still many technologies that I did not know, like SOLID, Microservice Architecture, and CQRS with MediatR pattern etc. According to [11] [8] CQRS (Command Query Responsibility Segregation) pattern, when combined with the MediatR library, is a software architectural approach used to enhance the scalability and maintainability of applications. It was extremely challenging for me to learn their entire tech stack because their projects are large, complex projects. So, I tried to learn as many of those as I could and focused on mastering their main technologies that were used frequently.

During the examination phase, I explored their projects' codes in Visual Studio to be familiar with the frontend and backend codes. I also looked at their tables in the database MS SQL Server. I tried to understand where different components of the projects were present. I also ran the projects to see what the client-side looked like, and also used a debugger to understand the flows of execution.

My industrial supervisor Tanvir Mahatab sir assigned me the task of creating Restful Web APIs in the Administration module. These APIs retrieve data from the database and then store new data in it. I was also instructed to check whether inserted data was successfully saved in the database. So I developed the APIs using ASP.NET Core in Visual Studio IDE and using the database Microsoft SQL Server. According to [4], it is important to learn LINQ to build the API. So, I learned CQRS with MediatR pattern, LINQ and OOP concepts to build the API.

4.2 Getting familiar with ASDBD's Application Architecture

After examining their project's contents in Visual Studio, I understood that ASDBD uses Microservices with Clean Architecture showed in (Figure:4.1). A clean architecture microservice (Figure:4.2) follows organized design principles, separating code parts for clarity and easy development and management in larger systems [9]. Scalability is a significant factor that influences the decision to adopt the microservices route.

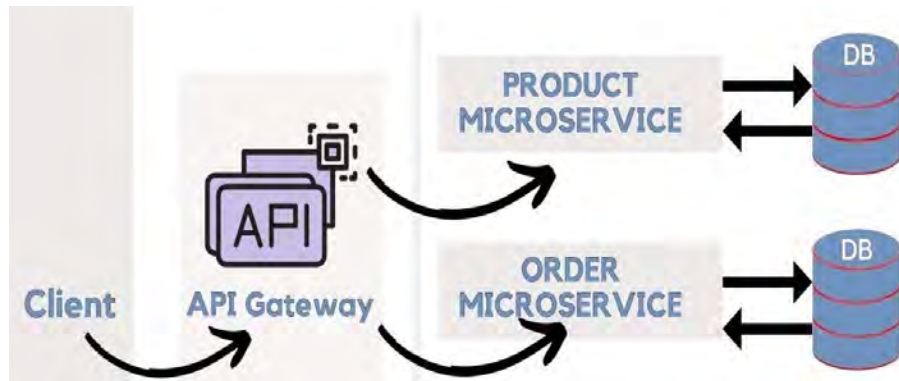


Figure 4.1: ASD Microservice Architecture

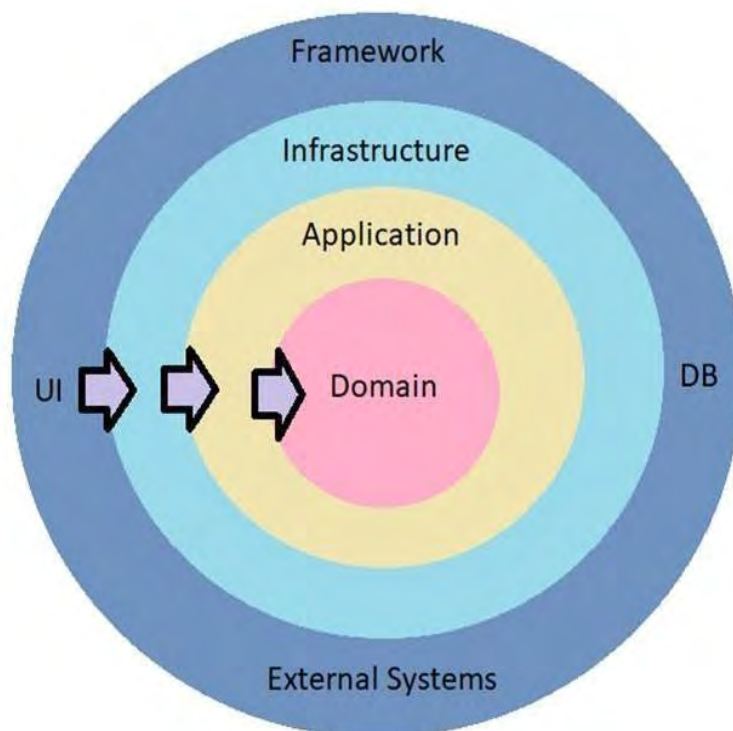


Figure 4.2: Microservices with Clean Architecture

The microservices architecture of ASD uses the concepts of MVC. The architecture utilizes only the 'M' (Model) and 'C' (Controller) concepts. However, for the 'V' (View), ASD employs Angular in the frontend. In this setup, the Model class represents the data, and the Controller class manages the communications between the Model and the View (frontend). Each class in a microservice is responsible for a specific business function or data access task. While the MVC concepts focus on the separation of concerns and user interface, microservice architecture concentrates on service-oriented architecture, scalability, and loose coupling. The main challenge with typical monolithic MVC apps lies in the number of dependencies, a concern that microservices aim to address by focusing solely on their own specific dependencies.

4.3 Environment Setup for Web API Development

The Following Tools and Software used for our Development:

1. Operating System: Windows
2. Editor: Visual Studio 2022
3. Dot Net Core SDK 3.1
4. Database: SQL Server 2019
5. SSMS (SQL Server Management Studio): To interact with SQL Server database
6. API Client: Postman, Swagger. It is used to test the Web APIs

4.4 .NET SDK

.NET SDK is the software development KIT which is helpful for the development and running of the application in the system. It's essential for developing and executing ASP.NET Core applications, encompassing the .NET runtime, libraries, and command-line utilities. I installed the .NET SDK from the official .NET website.

4.5 Package Manager

We use two popular package managers for our .NET Core development.

1. NuGet: By using NuGet we can manage and install packages, libraries, and dependencies in our ASP.NET Core project.
2. npm: As we are using front-end technology such as Angular in our ASP.NET Core project, we have to use npm (Node Package Manager) to manage our JavaScript packages and dependencies.

4.6 ASP.NET Core

By using this framework, we build our web applications including Web APIs. ASP.NET Core[5] is included in the .NET SDK which can be installed using the command line or through the Visual Studio installer.

4.7 Setup Visual Studio

The extensive features and integrations of Visual Studio make it an invaluable tool for software development [12]. First, I opened Visual Studio 2022 IDE. The project 'EnterpriseOfficeSuite' was selected from the list of projects shown in the IDE. Because the project is vast, it takes some time to load. Then, I opened the folder 'src' and then the folder 'Services' and finally the folder 'Administration'. API methods were expected to be created inside the 'Administration'.Api' project, so I entered it. The folder 'Controllers' contained the class 'BusinessProfileController', where I created the API methods.

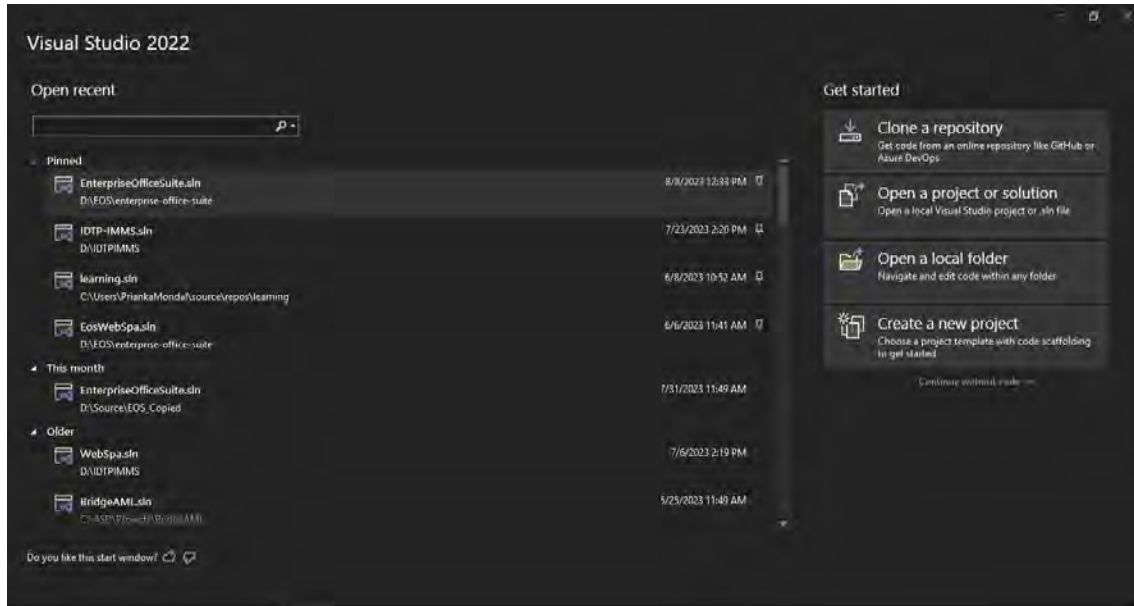


Figure 4.3: Visual Studio 2022 User Interface

I worked on the project 'EnterpriseOfficeSuite'. This project is composed of many compact, self-contained services including Administration, Ams, Dms, Crm, Hrm, Identity, Inventory, Ipoa, Msm, Ocm, Pms, Report, Sales, Wms, etc. For this reason, the architecture of the project is called Microservice Architecture. I basically worked on 'Administration' service, which is shown in (Figure 4.4).

The service 'Administration' is the backend and the application 'WebApps' is the frontend. 'Administration' gives us the API, and 'WebApps' exchanges and fetches data by communicating with the API. 'WebApps' displays the data in the frontend. Programs for communicating with the database are contained within the service.

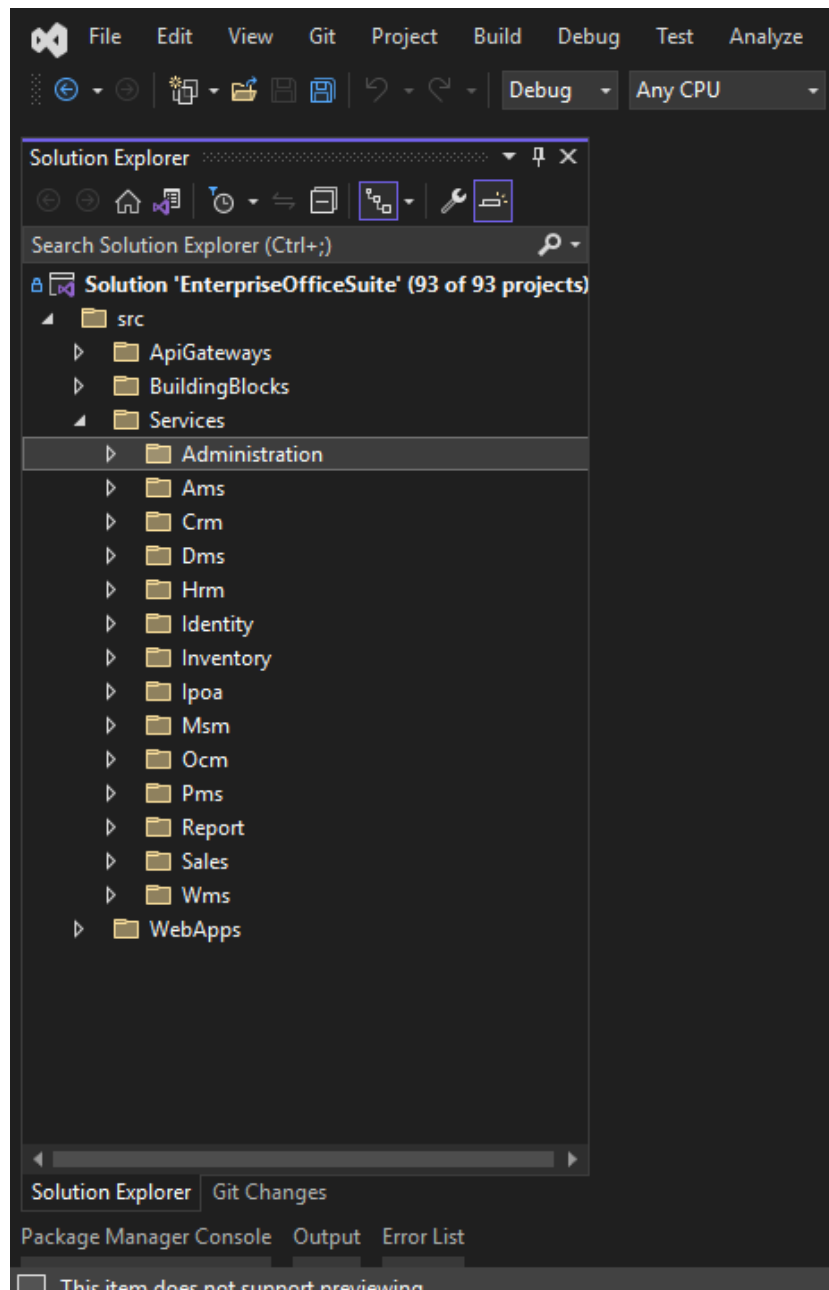


Figure 4.4: The Administration Folder

After opening the ‘Administration’ folder , we can see the API project named ‘Til.Eos.Administration.Api’. Sending data to and receiving data from the backend are done by this API project. ‘Controllers’ folder is like a brain that controls everything of this service. From where to take data, who will give data, from whom to receive data, all such backend related works are controlled in the ‘Controllers’ folder. Controllers folder contains logics of these things. Some configurations are required to exchange data with the database. Those configurations are in ‘Infrastructure’ folder. The ‘Core’ folder contains the Model, and the Controller requests data using the Model.

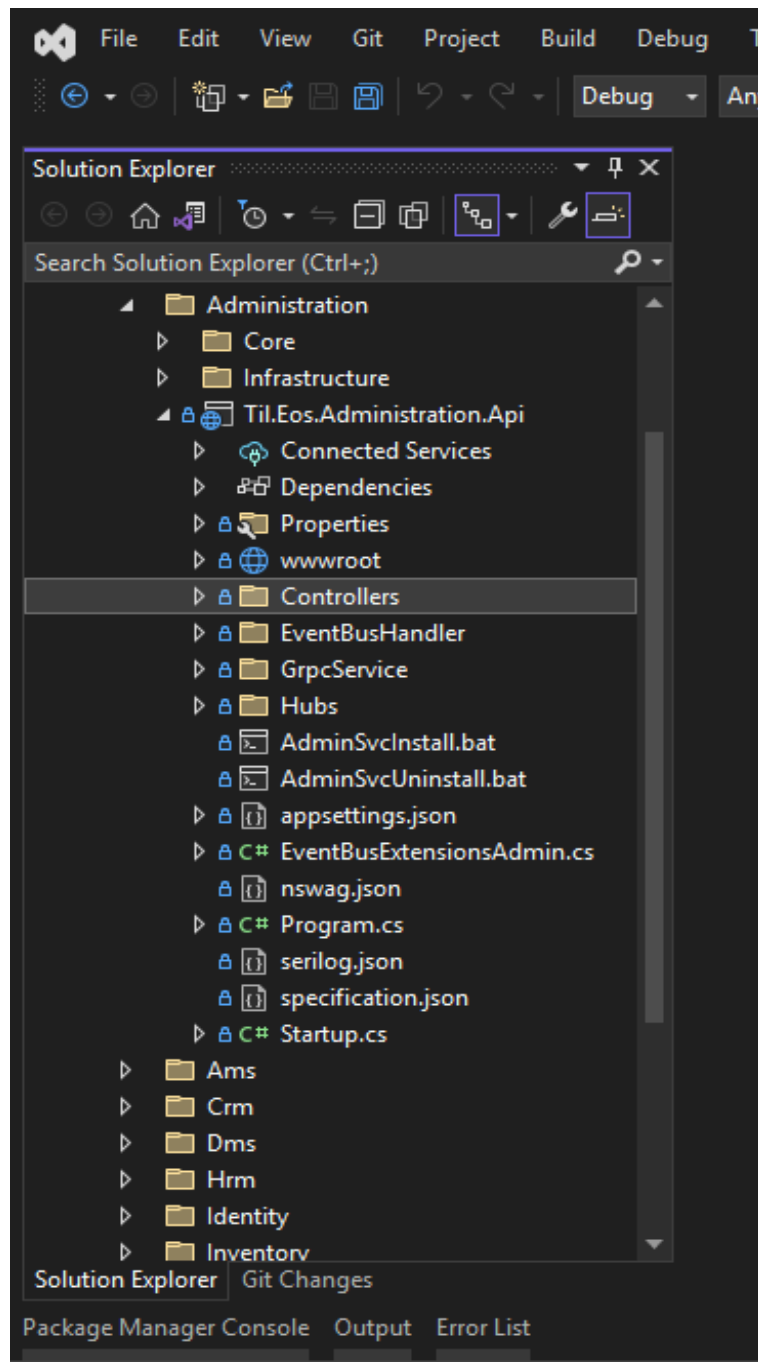


Figure 4.5: The Controllers Folder

In this 'Til.Eos.Administration.Api' project, various works are done with various data sources. That's why 'Controllers' folder contains different classes for different data sources and different purposes. My work was about business profile, so I worked on 'BusinessProfileController.cs' class.

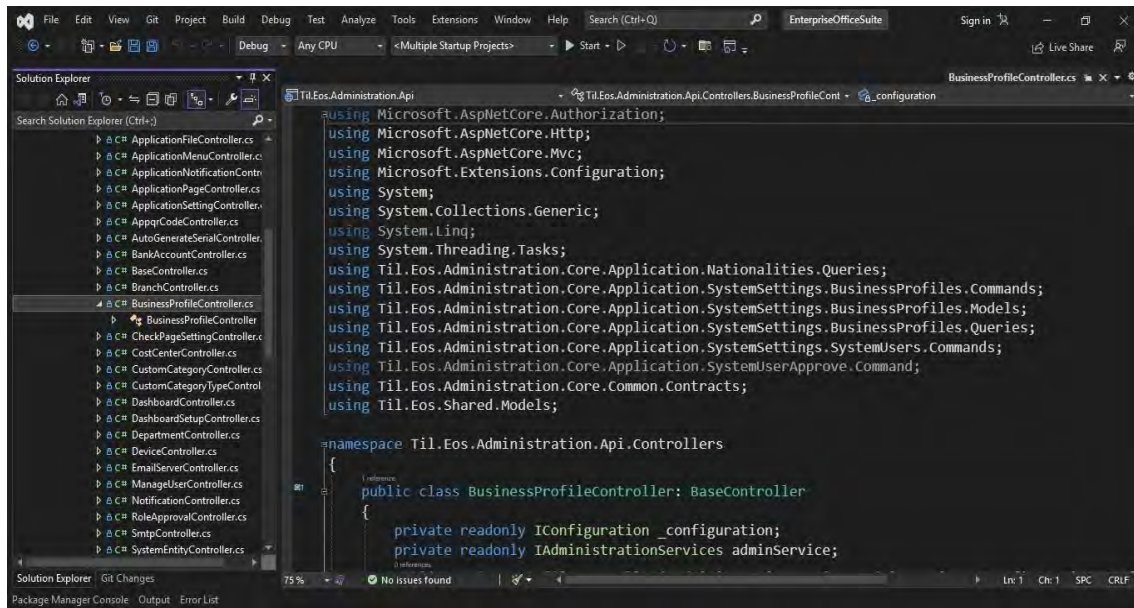


Figure 4.6: 'BusinessProfileController' Class

4.8 Building a Restful Web API

A Restful Web API takes inputs as needed and provides data as outputs to the users[6] [7]. The process of building it is actually easy and contains a few steps. Inside 'BusinessProfileController', I created the necessary API methods following the CQRS pattern. These methods use query/command classes, and a query/command class needs a view model. A view model is a class that contains the output data. First, I copied a view model class of another controller and pasted it in my desired folder. As needed, I renamed the file and its contents. Inside the view model class, I added necessary properties that I wanted in the output. Next, I copied a query/command class of another controller and pasted it in my desired folder. As needed, I renamed the file and its contents. Inside the query/command class, I added necessary inputs as properties and added the above-mentioned view model to hold output. Then, I added code for performing desired tasks inside a handler class, which is inside the command/query class. Finally, I created the API method by copying and pasting a similar method from any controller and renaming the pasted as necessary. Inside the method, I also added the query/command class mentioned above. Thus, I built Restful Web APIs.

When we want to get data from the database, we add the `HttpGet` attribute to the API method. Figure 4.7 shows how to generate a short code from a company's name. 'GetShortCodeQuery' class contains all the properties needed to obtain this functionality. Tanvir sir told me some criteria based on which each short code should be generated. For example, if the input name is "Advanced Software Development", then the short code 'ASD' should be created by combining the first three letters of each word of the full name. And, if there are two words, then the first letter of the first word and the first letter of the second word should be combined to make the short code. Again, if the company name is of one word, then only the first 10 letters should be taken as short code.

```
[HttpGet]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
public async Task<ActionResult<string>> GetShortCode(string name)
{
    var string query = await Mediator.Send(new GetShortCodeQuery { Name = name });
    return Ok(query);
}
```

Figure 4.7: Short Code Generating API

Then, Tanvir sir asked me to check whether the generated short code exists in the database or not. The API to verify a short code's existence is shown in Figure 4.8. This API will return boolean values. If the short code exists, true is returned. Else, false is returned.

```
[HttpGet]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
public async Task<ActionResult<bool>> ExistsShortCode(string shortCode)
{
    var bool query = await Mediator.Send(new ExistsShortCodeQuery { ShortCode = shortCode });
    return Ok(query);
}
```

Figure 4.8: API For Short Code Existence

If we want to add anything to the database, we have to use the `HttpPost` attribute in the API. Figure 4.9 shows how to create the API for an external sign up. When a user signs up, the user creates an account by adding various information such as name, date of birth, country, etc. Upon creation of the account, the data is added to the database. But, making any changes to the database requires a JWT token. Authorization is the main reason for using JWT tokens. My senior developers have already developed this 'BanglaMethod'. Calling this 'BanglaMethod' returns us a token as a string. This token is the permission to add something to the database. In this case, the user, who has signed up for his account right now, gets his sign-up information stored in the database.

```

[HttpPost]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesDefaultResponseType]
public async Task<ActionResult<UpsertResponse>> ExternalSignUp(CreateUserCommand command)
{
    DeviceController deviceController = new DeviceController(adminService, _configuration);
    string token = await deviceController.BanglaMethod();
    command.Token = "Bearer " + token;

    var UpsertResponse data = await Mediator.Send(command);
    return Ok(data);
}

```

Figure 4.9: User Sign Up API

After a user has signed up for his account, an email is sent to his email address to activate his account. Then, the work of email verification is completed by clicking the link sent to the email. As a result, the “Email Verified” column in the database gets set ‘true’. This email verification API is shown in Figure 4.10. At the time of sign up, an ID is assigned to the user. Using this ID, information related to the verified email address is added to the database. ‘BanglaMethod’ inside DeviceController generates the required token. This token is needed to get permission to add the verified email’s information to the database. We use ‘Guid.Parse(id)’ method to convert a string ID into a GUID and to store it in the database. We use GUID to ensure that database records are not duplicate, and each data has an ID that is unique across different databases. If the email verification information is successfully stored in the database, it can be known by the boolean value returned by the API method. If successful, it returns true. But, if it fails due to an internal server error, then it returns false.

```

[HttpPost]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesDefaultResponseType]
public async Task<ActionResult<bool>> VerifyUserEmail(string id)
{
    DeviceController deviceController = new DeviceController(adminService, _configuration);
    string token = await deviceController.BanglaMethod();

    SystemUserEmailVerifiedCommand command = new SystemUserEmailVerifiedCommand();
    command.Token = "Bearer " + token;
    command.Id = Guid.Parse(id);
    bool data = await Mediator.Send(command);
    return Ok(data);
}

```

Figure 4.10: Email Verification API

In the database, there is a table named 'Nationalities', which contains the names of all nationalities in the world. Using this table, the GetNationalitiesList API returns all nationality names in the form of a list. Each item of the returned list is a 'SelectModel', which contains each nationality name. Figure 4.11 shows how the list of all nationality names can be collected using theHttpGet attribute.

```
[HttpGet]
[ProducesResponseType(StatusCodes.Status200OK)]
public async Task<ActionResult<List<SelectModel>>> GetNationalitiesList()
{
    var GetNationalitiesSelectListQuery query = new GetNationalitiesSelectListQuery();
    return Ok(await Mediator.Send(query));
}
```

Figure 4.11: Nationality List Collection API

4.9 Swagger API

Swagger API is used to test whether the API methods work and display outputs as expected [10]. I created only GET and POST methods. For instance, I created a GET method that collected the names of all nationalities from the database and displayed the list as output. An instance of a POST method I created was for user Sign Up. User's information was collected by the API and stored in the database. These API methods collect and also output data in JSON format.

To launch the Swagger page, I clicked the 'Start' button on Visual Studio with the 'Administration.Api' project selected for running. This launched Swagger in the browser. On that browser-page, I looked for the section 'BusinessProfile' because it contained the API methods for 'BusinessProfileController'. For testing any method, I clicked its name and entered the necessary input. Thus, Swagger displayed outputs upon executing the method.

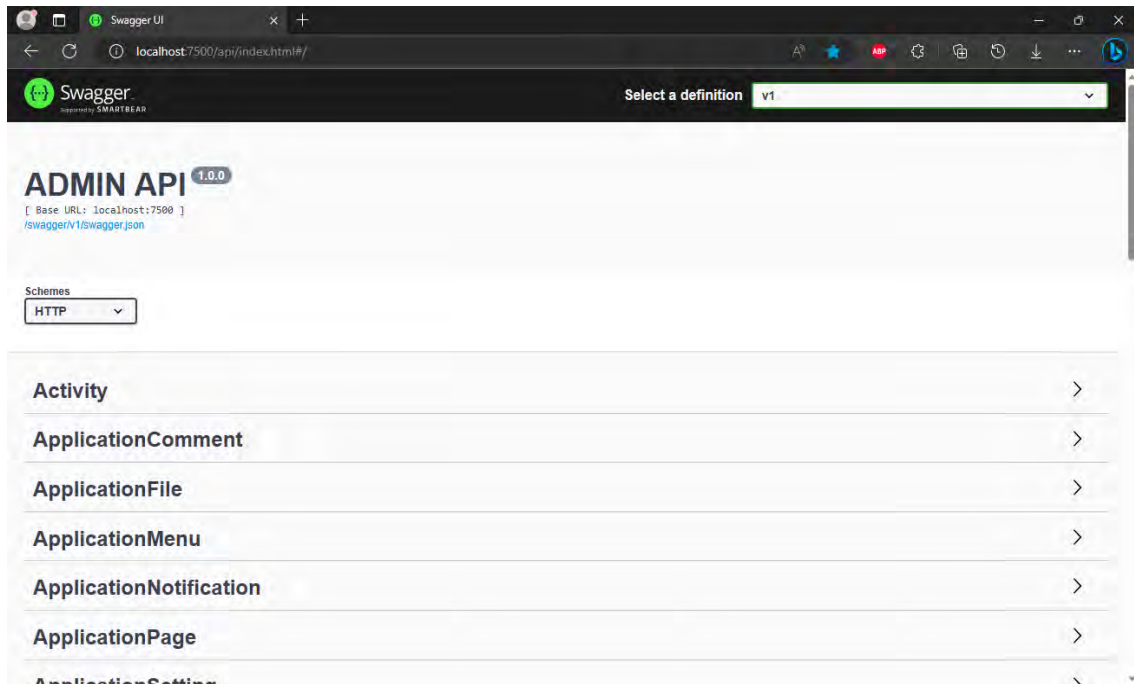


Figure 4.12: Swagger API Page

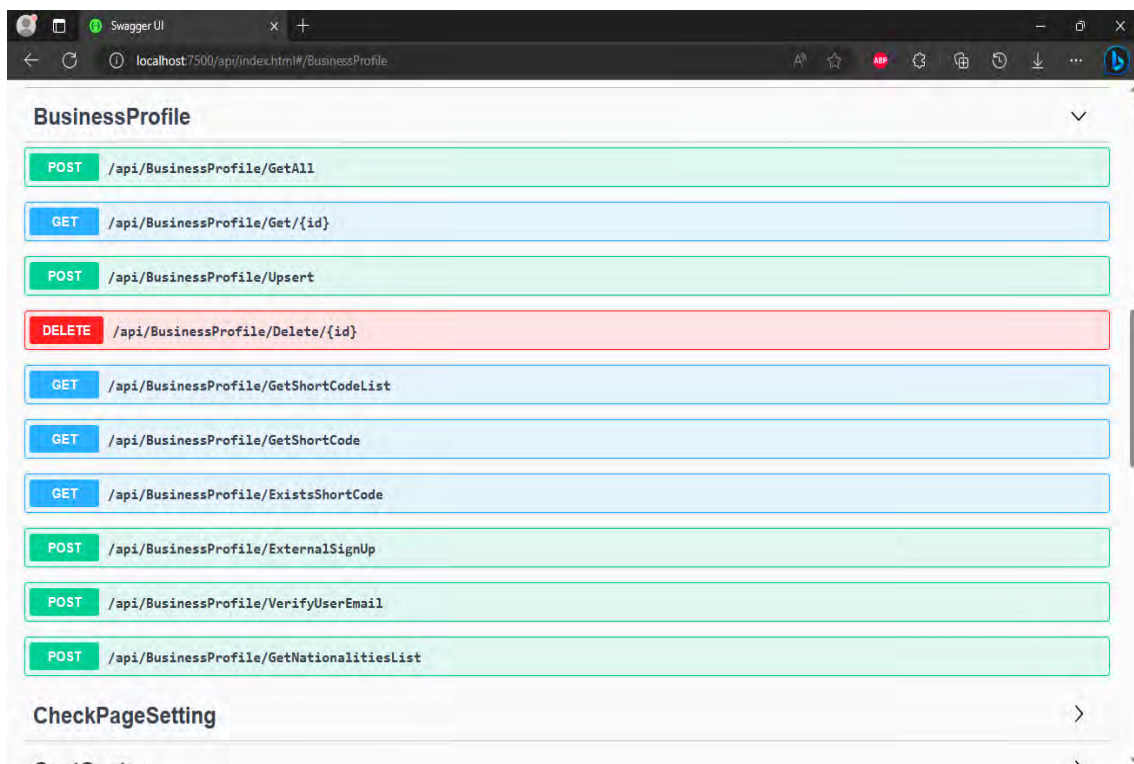


Figure 4.13: BusinessProfile On Swagger

4.10 Postman

Postman was used to check whether the API endpoints work properly. After writing code for the APIs, I performed tests with Postman. Most of the time, these API tests were successful in Postman. However, sometimes I encountered failing test results and made necessary corrections to the API code. In all the following figures, API tests were successful, and the success status (200 OK) code was displayed.

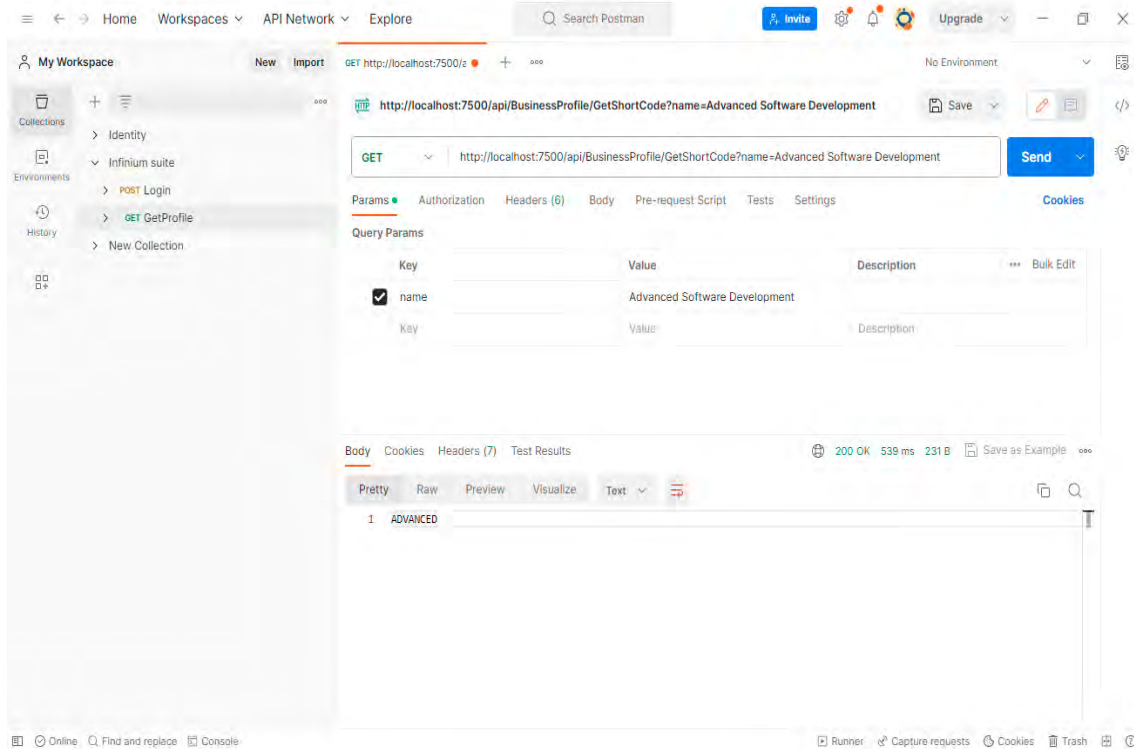


Figure 4.14: Postman Testing For Short Code Generating API

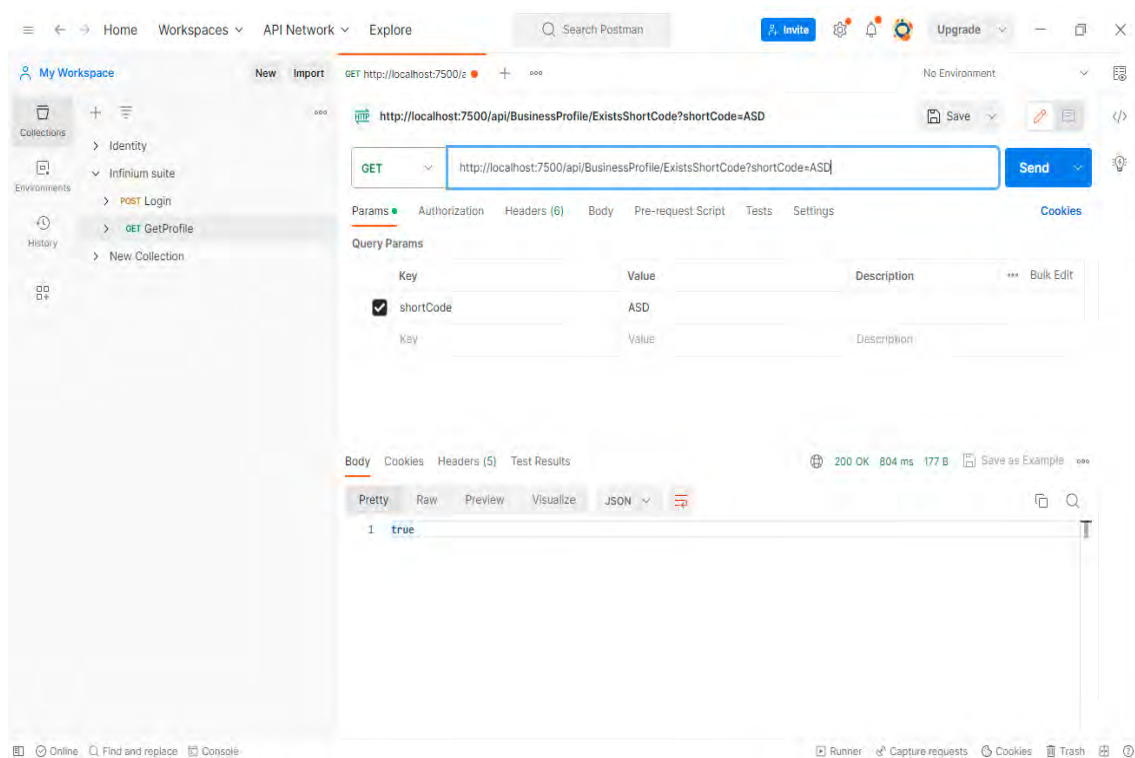


Figure 4.15: Postman Validation Of API For Short Code Existence

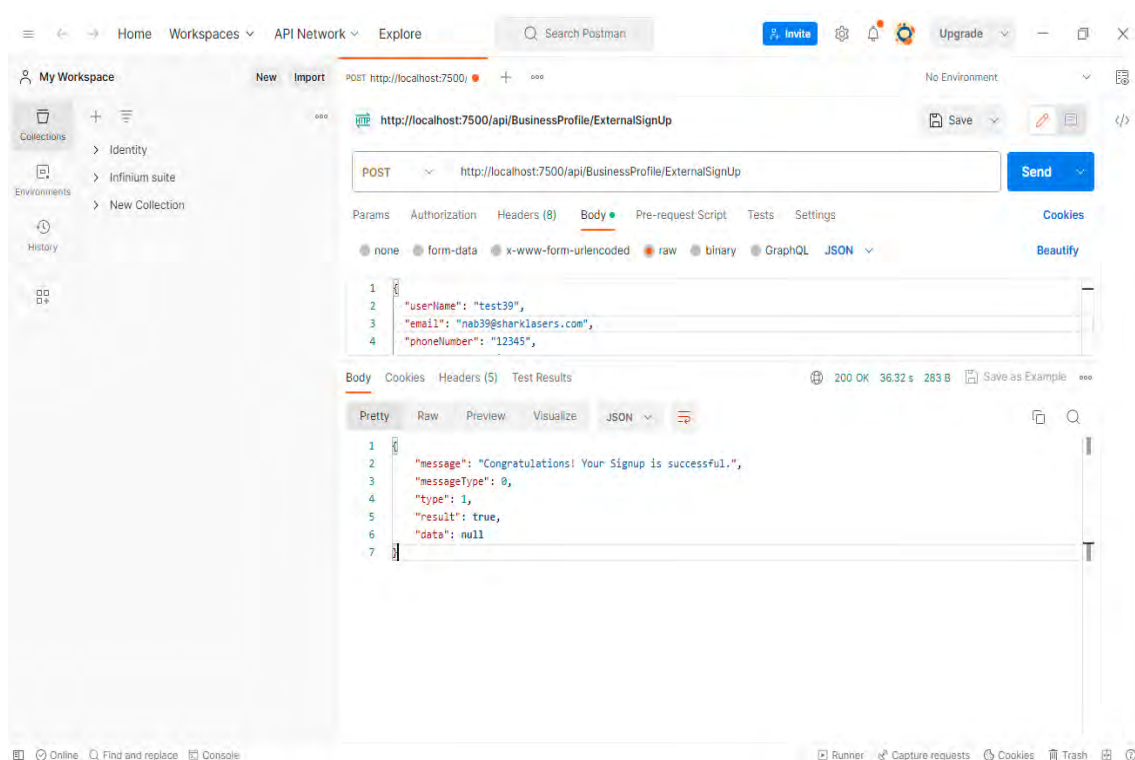


Figure 4.16: Postman Testing - User Sign Up API

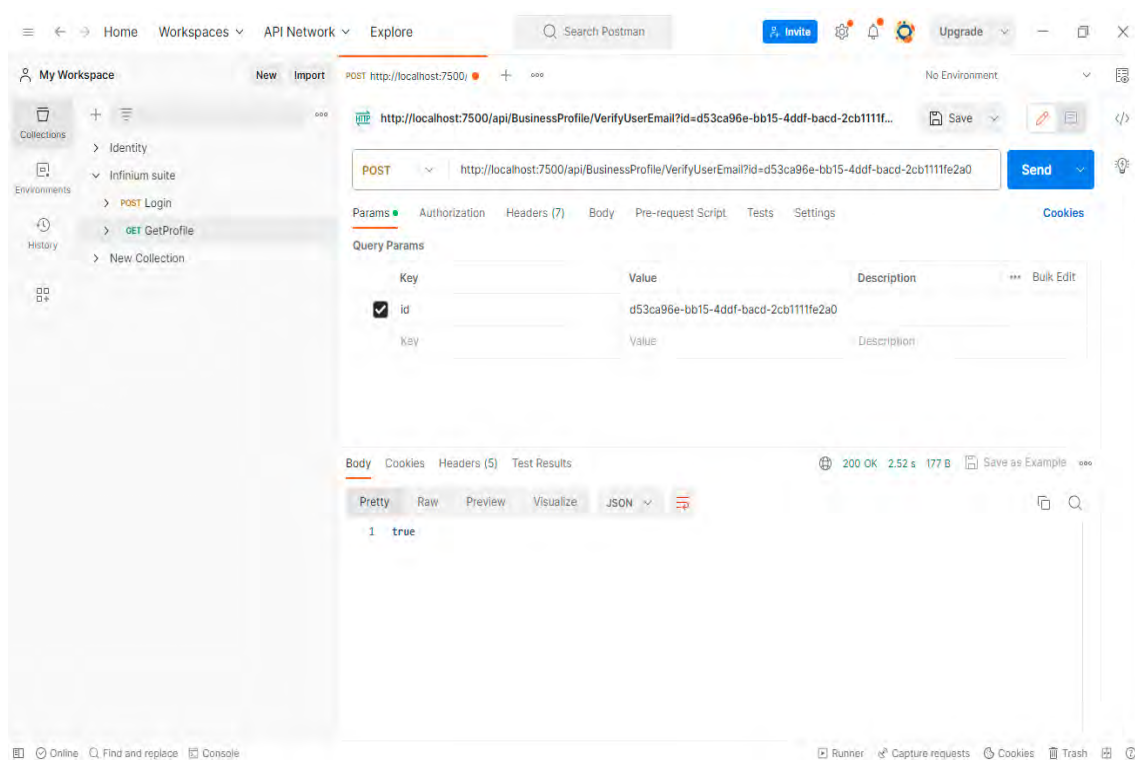


Figure 4.17: Postman Testing - Email Verification API

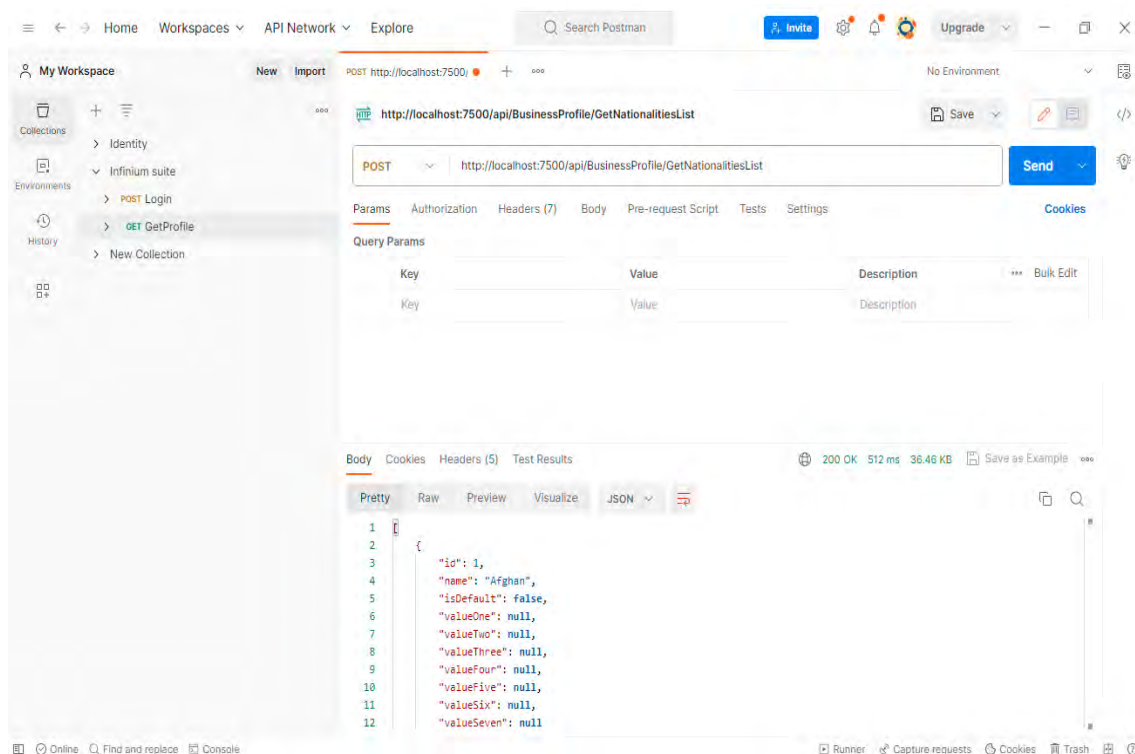


Figure 4.18: Postman Testing - API for Nationality List Collection

4.11 Postman vs. Swagger: Uncovered

After using both Postman and Swagger, I figured out how they're different. At first, I thought they were kind of the same, both used for testing APIs, but turns out, they have their own things. Swagger is like a blueprint or a plan for building something. It helps decide how things should look and work. In the world of computers and apps, Swagger is used early on to plan how an application communicates with other applications, such as figuring out how a phone app should get information from a server. On the contrary, Postman plays the role of a tester, ensuring that everything operates as intended. After the plan is crafted with Swagger, Postman steps in to verify the smooth operation of different parts of the app by sending messages and checking their responses.

4.12 Microsoft SQL Server

I opened the database Microsoft SQL Server Management Studio to check existing data and to check whether new data insertion operation worked as expected.

I opened the database on the computer and clicked the 'connect' button with the Login name and password already present there. Then, the server was loaded in the 'Object Explorer' of the database. I clicked to open 'Databases' and then opened the database that contained my desired table. Then, I clicked to open 'Tables' and located the table 'system.BusinessProfile'. I right-clicked the table-name to open a menu, and in the menu, I clicked 'Select Top 1000 Rows'. Thus, data contained in the table became visible.

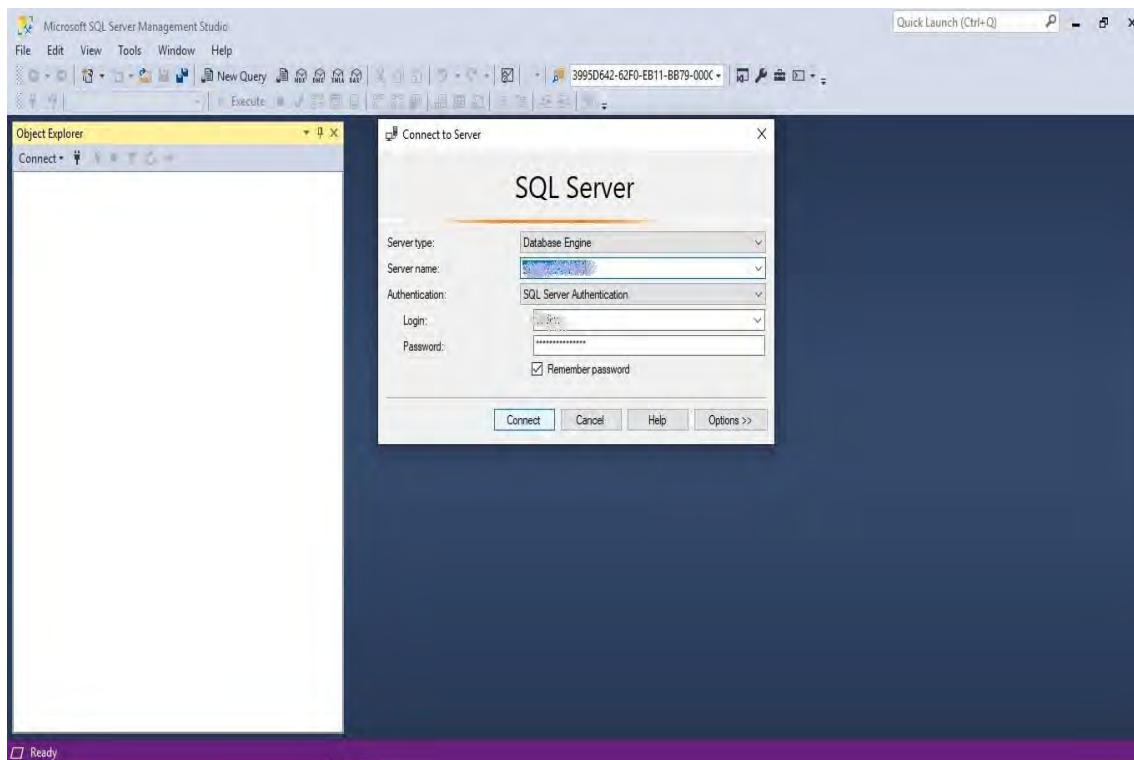


Figure 4.19: Microsoft SQL Server Management Studio

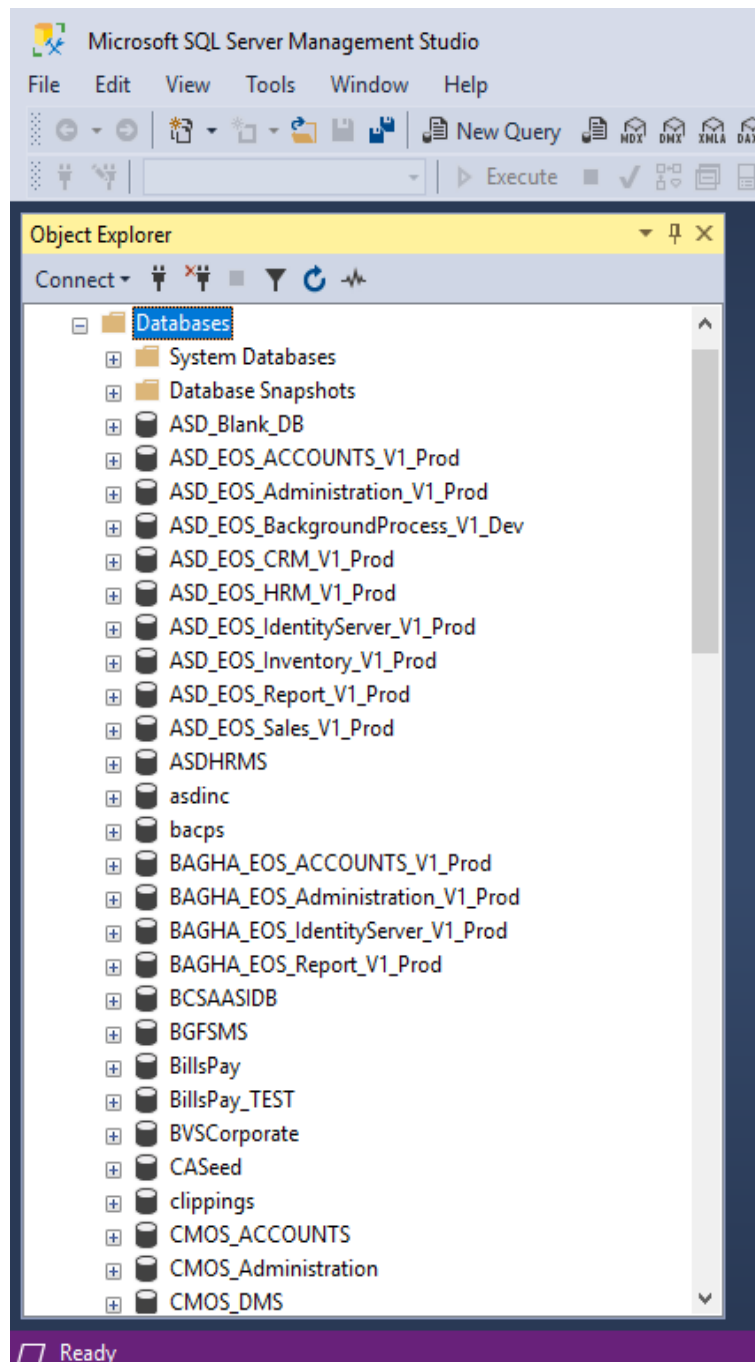


Figure 4.20: Databases Folder

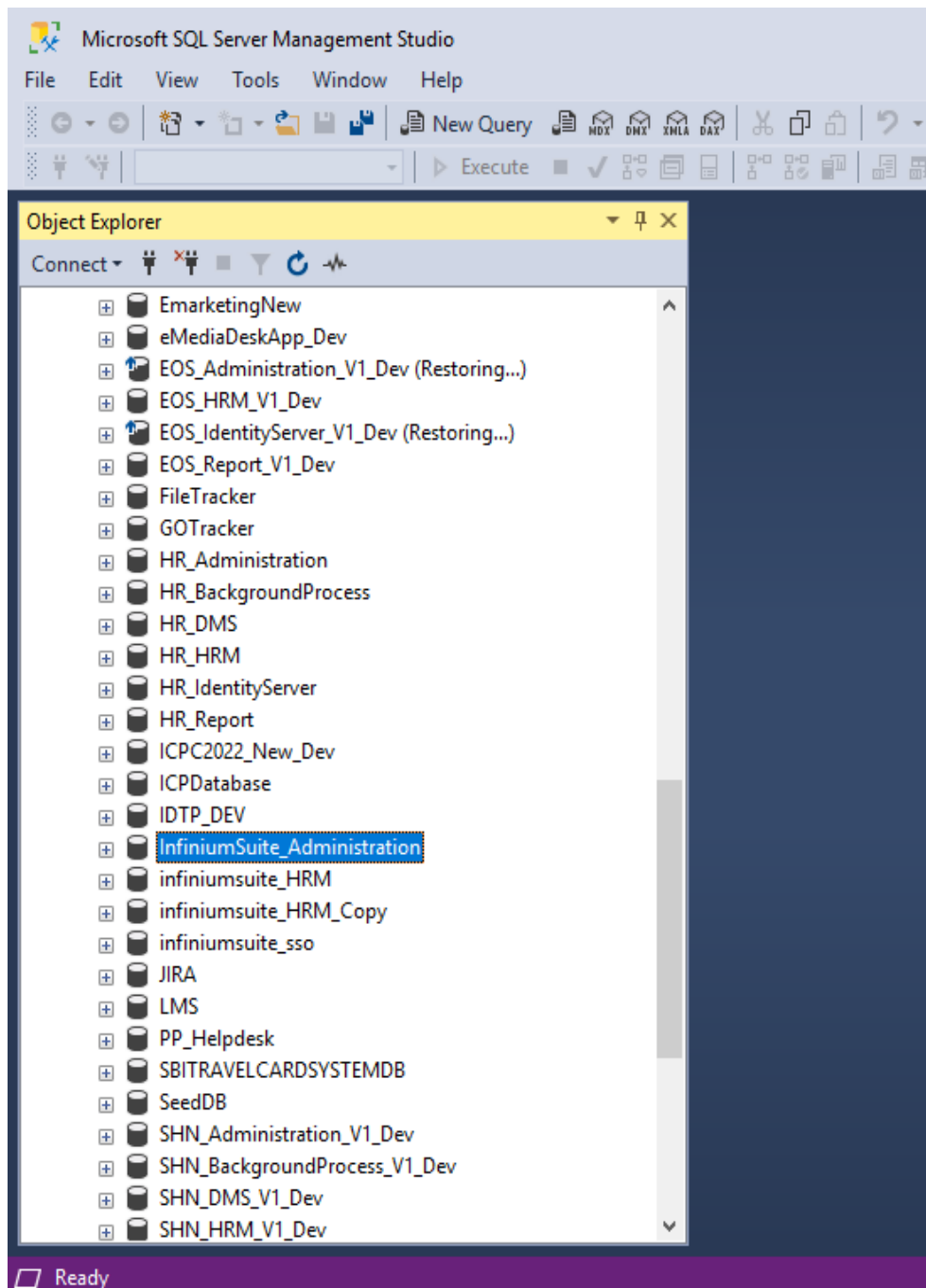


Figure 4.21: InfiniumSuit_Administartion Folder

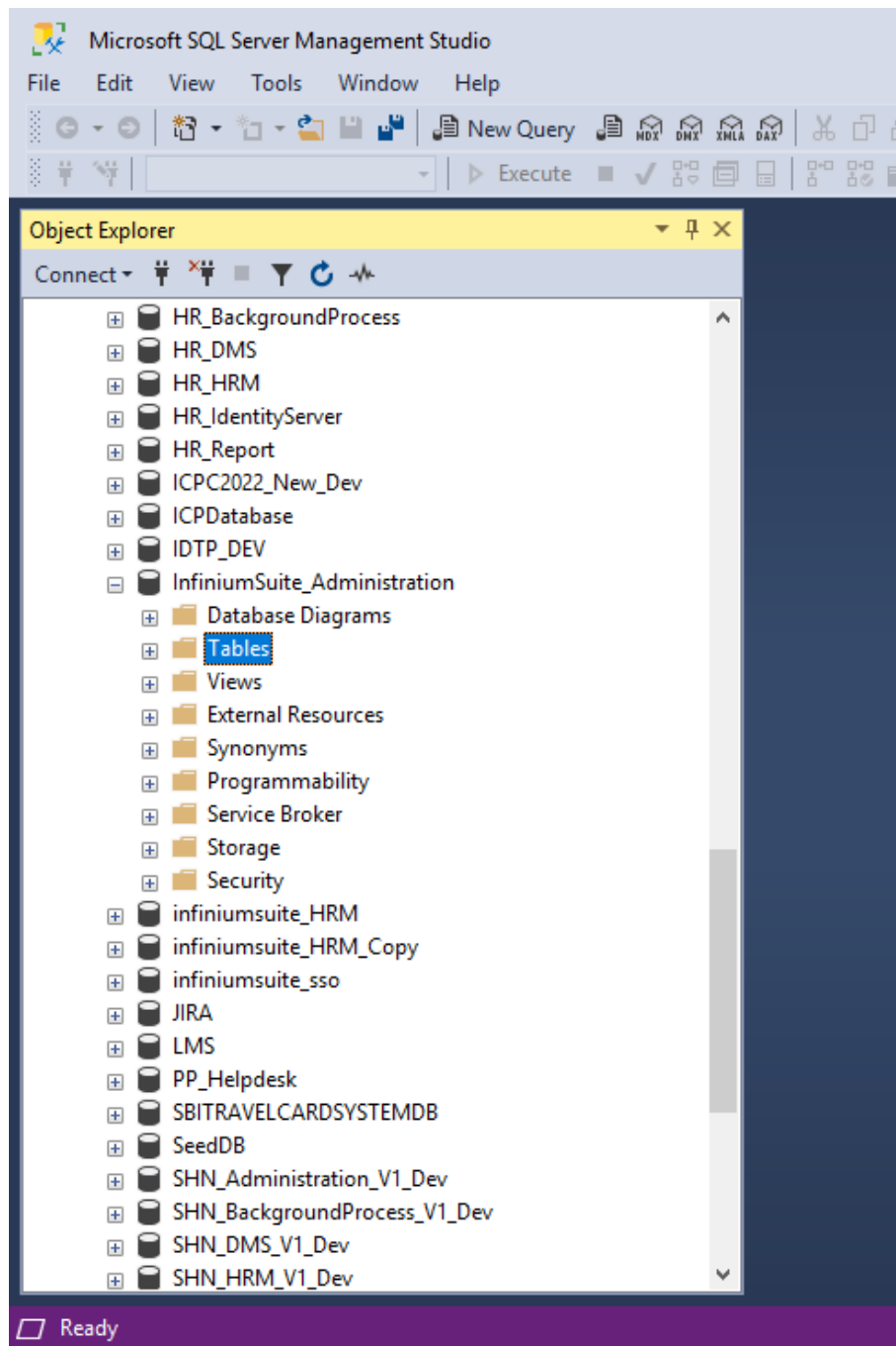


Figure 4.22: Tables Folder

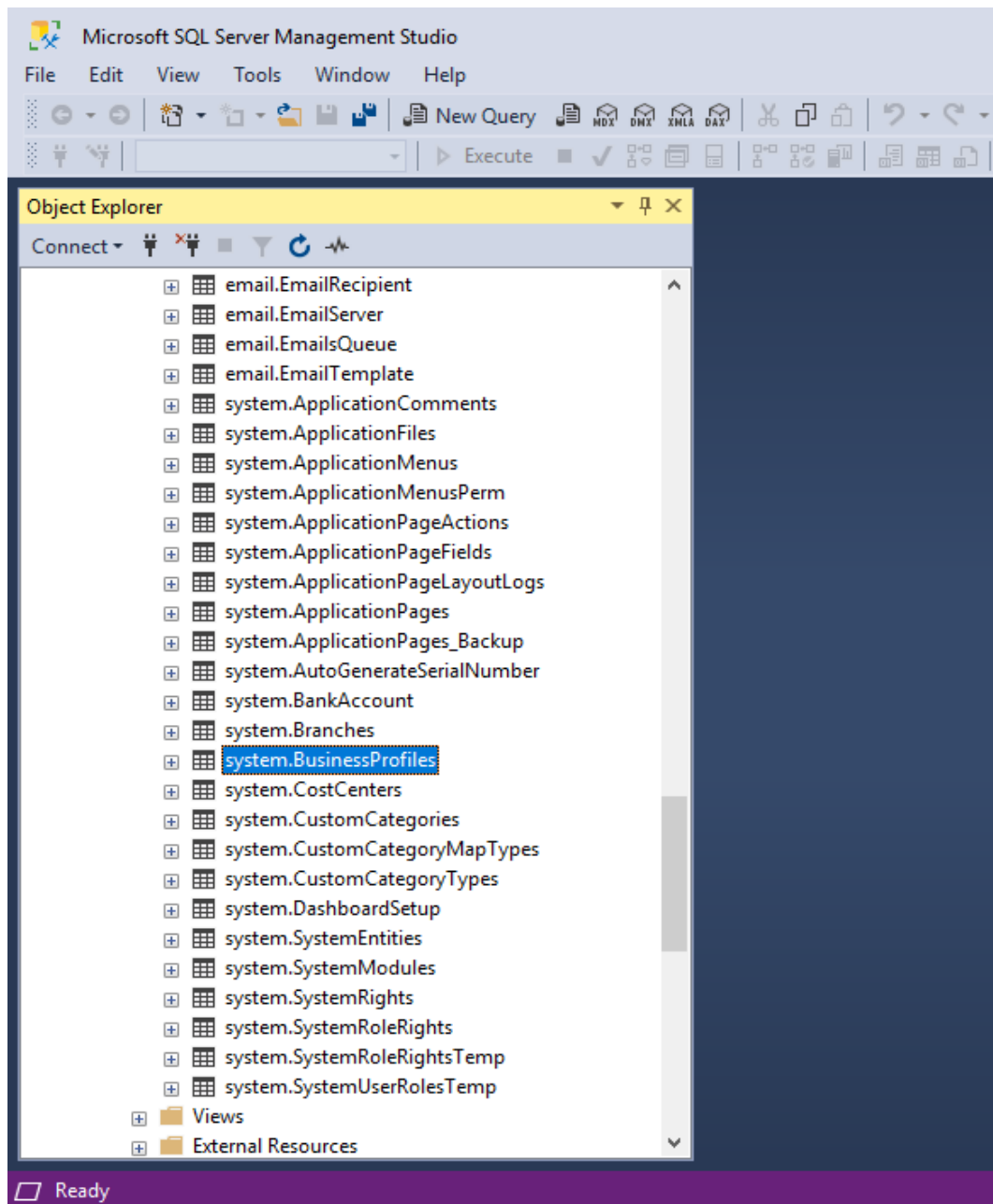


Figure 4.23: system.BusinessProfiles Folder

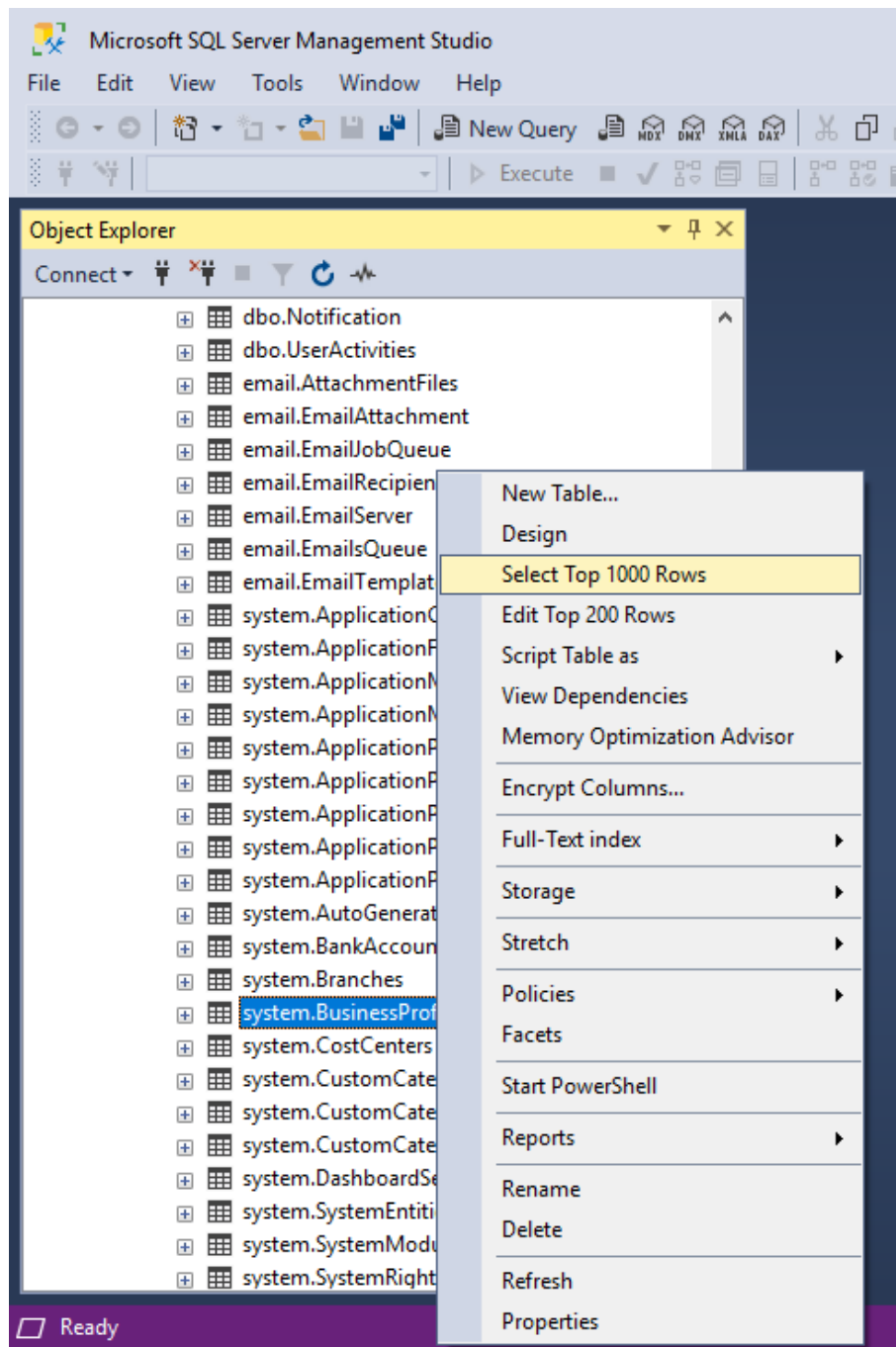


Figure 4.24: Selecting Top 1000 Rows

4.13 Tutorials and Assistance

I utilized online resources to supplement my learning. Specifically, I followed YouTube tutorials by Mosh Hamedani and Kudvenkat to learn C sharp programming and object-oriented programming (OOP). Furthermore, my industrial supervisor Tanvir Mahatab sir recommended me to follow Microsoft Documentation for learning Microsoft SQL Server and for Restful Web API, he provided me to take the courses titled “Restful Web API—The Complete Guide (.NET7 API) 2022-8” created by Bhrujen Patel on Udemy.

4.14 Learning Path

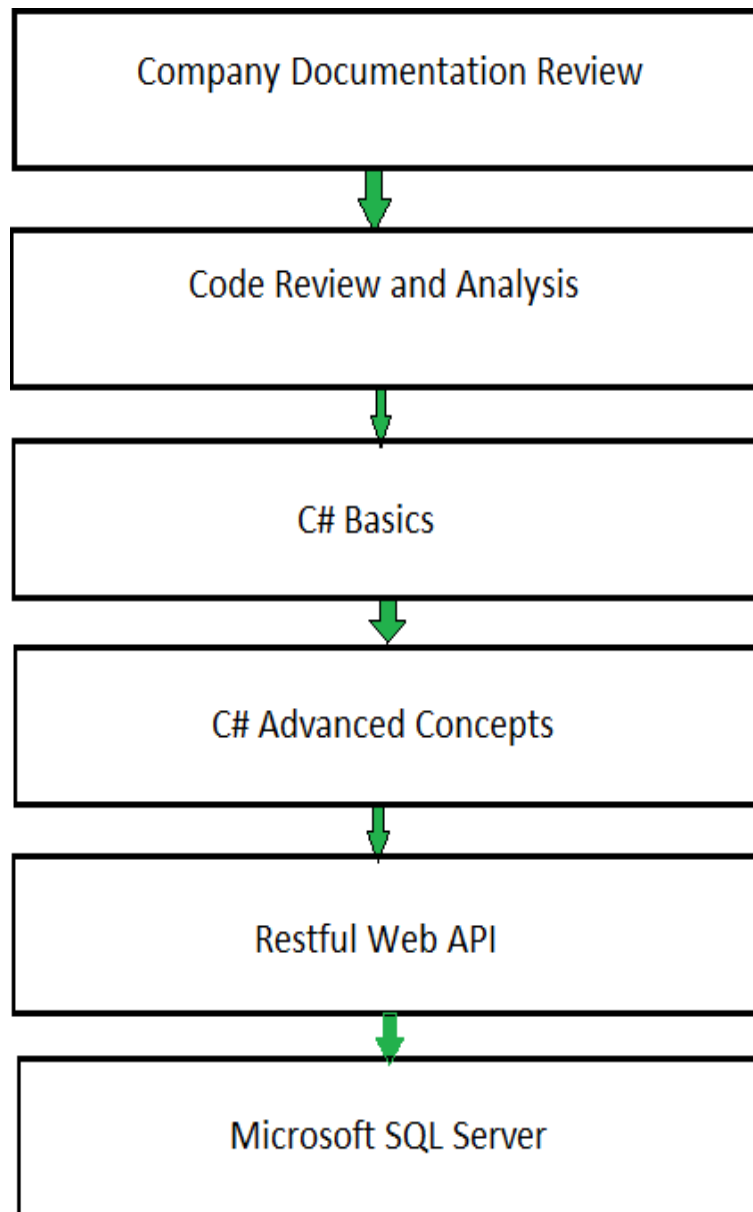


Figure 4.25: My Learning Path

Chapter 5

Conclusion

5.1 Summary

In conclusion, my exciting six months at ASDBD has given me a valuable peek into software development. I started this journey on June 1, 2023, and it's a six-month journey, ended on November 30, 2023. Although I began as a newcomer, I tackled tasks that professionals handle, and this hands-on experience greatly contributed to my learning journey. I dived into things like RESTful Web APIs, Microsoft SQL Server, and Swagger, and during this time, I could see a big improvement in what I understood. Notably, a significant portion of my internship was dedicated to mastering Postman, this tool that helps test and work on APIs with others. It's been like learning while doing things, and now I get why these tools are so important. As my internship went on, I took the chance to learn more about something called microservices. I faced some challenges and had some limitations. My seniors advised me to focus on learning first before hands on work. There were some tough parts, but each one was a chance for me to get better at what I am doing.

5.2 Future Work

I recently completed my internship as a Software Engineer trainee, but my eyes are still set on a future role as a Data Engineer. I had my sights on this field because it's in high demand in the tech industry. During my internship, I built on the knowledge learned from the Database Systems course and enhanced my SQL skills, which is like the language for talking to databases. The cool part is that what I learned in software development is going to be super handy for my future as a Data Engineer. Making software taught me how to solve problems and design things efficiently. These skills are like secret weapons that I can use when working with data. So, my internship wasn't just about software; it was a stepping stone to a future where I get to play with big data and help businesses make smart decisions.

Bibliography

- [1] H. Jain, P. Vitharana, and F. M. Zahedi, “An assessment model for requirements identification in component-based software development,” *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, vol. 34, no. 4, pp. 48–63, 2003.
- [2] K. Schwaber, *Agile project management with Scrum*. Microsoft press, 2004.
- [3] K. Schwalbe, *Introduction to project management*. Course Technology Cengage Learning Boston, 2009.
- [4] M. MacDonald and M. MacDonald, “Linq and the entity framework,” *Beginning ASP. NET 4.5 in C#*, pp. 753–790, 2012.
- [5] J. Kanjilal, *ASP. NET Web API: Build RESTful web applications and services on the. NET framework*. Packt Publishing Ltd, 2013.
- [6] A. Del Sole and A. Del Sole, “Creating restful web api services,” *Beginning Visual Studio for Mac: Build Cross-Platform Apps with Xamarin and. NET Core*, pp. 261–281, 2017.
- [7] S. Resca, *Hands-On RESTful Web Services with ASP. NET Core 3: Design production-ready, testable, and flexible RESTful APIs for web applications and microservices*. Packt Publishing Ltd, 2019.
- [8] C.-H. Marcotte, *An Atypical ASP. NET Core 5 Design Patterns Guide: A SOLID adventure into architectural principles, design patterns,. NET 5, and C*. Packt Publishing Ltd, 2020.
- [9] C. Rajasekharaiah and C. Rajasekharaiah, “Microservices: What, why, and how?” *Cloud-Based Microservices: Techniques, Challenges, and Solutions*, pp. 13–40, 2021.
- [10] A. Golmohammadi, M. Zhang, and A. Arcuri, “Testing restful apis: A survey,” *ACM Transactions on Software Engineering and Methodology*, 2022.
- [11] O. Jejić, M. Škembarević, and S. Babarogić, “Defining software architecture modalities based on event sourcing architecture pattern,” in *European Conference on Advances in Databases and Information Systems*, Springer, 2022, pp. 450–458.
- [12] I. Dominte, “Setting up the environment,” in *Web API Development for the Absolute Beginner: A Step-by-step Approach to Learning the Fundamentals of Web API Development with. NET 7*, Springer, 2023, pp. 51–73.