201901306_Final_Submission_Plag

by Raj Patel

Submission date: 07-May-2023 02:37PM (UTC+0530)
Submission ID: 2086353096
File name: BTP_Report.pdf (88.39K)
Word count: 3327
Character count: 19556

MigrationWiz - Teams Private Chat Feature Development

Student Name: Raj Patel
Enrollment ID: **201901306**
B. Tech. Project (BTP) Report
BTP Mode: ***Off Campus***
Dhirubhai Ambani Institute of ICT (DA-IICT)
Gandhinagar, India
201901306@daiict.ac.in

Mentor's Name: Deep Dave
Company's Name: *Clomotech*
Company's Address: HD - 128, WeWork Krishe Emerald,
Kondapur Main Road, Laxmi Cyber City, Whitefields,
Kondapur, Hyderabad, Telangana 500081
deepdave@clomotech.com
On-Campus Mentor: BTP Coordinator

***Abstract*—This report outlines my industrial learning and involvement in the development of the MigrationWiz software project during my BTP under the guidance of my off - campus mentor Deep Dave. My assigned task was to implement various features in the application to fulfill its specific requirements and to research some of the features developed in the old related projects. Throughout my internship, I focused on both front-end and back-end aspects. On the front-end side, I handled tasks such as retrieving and processing data, making API calls, modifying the user interface and displaying relevant information. On the back-end side, I worked on writing queries for fetching and filtering the data from the database. This report offers an introductory overview of the software product I was involved in and showcases the essential insights, skills, and knowledge I gained throughout the development phase. It emphasizes my significant contributions to the project's advancement and highlights the outcomes achieved.**

***Index Terms*—Migration, Database, C#, Front-end, Back-end, Git, Azure, DevOps, UI, EmberJs, API, Query**

I. Introduction

MigrationWiz [1] is a software-as-a-service (SaaS) offering, designed to facilitate efficient migration of workloads across different domains. This comprehensive solution enables the seamless transfer of files, emails, calendars, documents, messages, chats, and more. MigrationWiz stands as a versatile tool capable of managing workload migrations across multiple cloud platforms. Its efficiency lies in its ability to handle data and email migrations from diverse sources and destinations. By utilizing MigrationWiz, users can effortlessly migrate emails, personal archives, documents, files, public folders, and even Microsoft Teams data, including channels, conversations, and files, from any source to any destination. To illustrate its practicality, consider a scenario where Company 'A' undergoes acquisition by Company 'B'. Prior to the acquisition, Company A's employees had their emails, chats, and documents stored in the @a.com domain. However, following the acquisition, the domain changes to @b.com. In order to gain access to their valuable documents, emails, and chats, a transfer to the @b.com domain becomes imperative. MigrationWiz serves as the ideal solution for seamlessly migrating and transferring these critical assets from one domain to another.

The initial functionality offered by the product entailed transferring 1:1 chats, group chats, and meeting chats as emails to the target domain. The documents from the source domain were migrated to the destination user's OneDrive account. These attachments from OneDrive were subsequently linked within the emails, resulting in the representation of each chat from the source domain as an individual email during the early stages.

The updated requirement called for a direct transfer or migration of chats to the Microsoft Teams destination, referred to as"Rehydration", in order to enhance the user experience. To address this need, a new project named "Teams Private Chat (Rehydration)" was initiated under the MigrationWiz platform, and I actively contributed as a member of the development team. My responsibilities encompassed working on both the front-end and back-end aspects of the project, implementing necessary features, and resolving any encountered bugs during the development process.

II. Learning Outcomes

During the course of the BTP, development period of the project, I acquired many technical and non - technical skills. I had the opportunity to familiarize myself with a wide array of industry standards, tools, and technologies. The invaluable experiences and competencies acquired throughout this endeavor are outlined below:

A. Tools and Technologies Learned

I had the privilege of gaining hands - on experience with a diverse range of cutting - edge tools and technologies, which greatly facilitate and enhance the development process for developers. This project served as an invaluable learning experience, teaching me the essential skill of discerning and effectively utilizing the most appropriate tools tailored to the project's requirements, amidst the vast array of available options. The following enumeration outlines the specific technologies that I acquired proficiency in during this period:

1) *Javascript:* Within the front-end codebase, a segment relied upon the programming language JavaScript, an influential and widely adopted scripting language employed by numerous web browsers. Through this learning experience, I gained insight into the language's integral role in enabling dynamic content alteration on user interface (UI) pages. JavaScript proved instrumental in orchestrating the logic-driven architecture of the code, primarily involving the dynamic manipulation of UI elements using the Document Object Model (DOM) provided by JavaScript. Additionally, I harnessed JavaScript's capabilities to author API calls and perform mathematical operations, further augmenting the functionality and interactivity of the application.

2) *EmberJs:* The front - end component of the project employed the EmberJs framework, which leverages the Model View Controller (MVC) architectural pattern. EmberJs is a JavaScript framework specifically designed to facilitate the development of large - scale client - side applications with relative ease. Compared to other MVC JavaScript frameworks, EmberJs promotes the creation of structured and organized code. It encompasses distinct elements such as Models, Controllers, Templates, and Routes, effectively dividing the framework into three layers [2]. The Controller Layer, consisting of Routes and Controllers, primarily focuses on updating the application's URL. The View Layer, comprising templates and views, assumes responsibility for dynamically updating the user interface with various elements. Lastly, the Model Layer, incorporating models and data, handles data formatting and facilitates server-side communications.

3) *C#:* To facilitate modifications in the project's back-end portion, I embarked on acquiring proficiency in the C# programming language. C# stands as a highly favored, contemporary programming language extensively employed by software developers, particularly in the realm of software product development. Based on the core tenets of Object-Oriented Programming (OOP), C# proved instrumental in broadening my skill set. I am delighted to have had the opportunity to delve into this widely acclaimed programming language, renowned for its versatility in constructing a wide spectrum of applications on the .NET platform. Throughout this endeavor, I familiarized myself with various imperative programming concepts inherent to C#, including Dependency Injection, Interfaces and Inheritance. Additionally, I gained invaluable insight into one of its most noteworthy and advantageous features, namely Language-Integrated Query (LINQ). I discovered that this distinctive capability seamlessly integrates the functionalities of database querying directly within the programming language, negating the necessity to acquire proficiency in specialized database query processing tools.

4) *AzureDevOps:* Throughout the duration of the BTP, I acquired comprehensive knowledge regarding the intricate process involved in software project development, encompassing meticulous planning, meticulous design preparation, coding, testing, continuous integration, continuous deployment, and subsequent maintenance. Employing disparate tools for each development phase often proved cumbersome and bewildering. Consequently, I was introduced to an integrated tool, AzureDevOps, which seamlessly manages all aspects of the development lifecycle. This cohesive solution offers exceptional and efficient features to streamline the entire software development process. Notably, Azure boards facilitated agile planning and task tracking, while azure pipelines seamlessly facilitated continuous integration (CI) and deployment (CD) processes. Moreover, azure repos served as a dependable platform for hosting source code, and azure test plans provided an integrated environment for rigorous code testing.

5) *Git:* Ensuring meticulous maintenance and preserving an accurate version history of the code constitute pivotal aspects within the software development lifecycle. To facilitate seamless tracking of changes implemented by multiple developers collaborating on a shared project, the adoption of a proficient version control system assumes paramount importance. Git, widely favored by developers, stands out as an exemplary choice for effective version control. During my involvement, I utilized Git to access prior iterations of the codebase, aligning with the specific task requirements. By seamlessly transitioning to designated branches, I could readily access earlier code versions. Furthermore, while committing my modifications, I encountered instances of merge conflicts arising from divergent code contributions by fellow developers. Consequently, I garnered valuable experience in resolving such conflicts and adeptly retaining the desired changes.

6) *VS Code and Visual Studio:* During BTP, I utilized two powerful integrated development environments (IDEs) – VS Code and Visual Studio. VS Code, a lightweight cross - platform code editor, provided an efficient and customizable environment for streamlined coding tasks. Visual Studio, a comprehensive IDE for Windows, offered advanced debugging, testing, and design tools, enhancing my productivity and software development skills. I particularly appreciated the seamless integration of Git version control within both IDEs, allowing for efficient code collaboration and easy management of project repositories. These IDEs played a significant role in my BTP, empowering me to code with precision and efficiency. Their comprehensive toolsets and Git

integration greatly enhanced my productivity and contributed to the success of my BTP.

B. Effective Teamwork and Communication

In the context of my BTP project, effective teamwork and communication played a pivotal role in achieving successful outcomes. Collaborating closely with my colleagues fostered a cohesive work environment, enabling the seamless sharing of ideas, knowledge, and responsibilities. I actively engaged in collaborative efforts with my team leader, ensuring clear and efficient communication channels. The code review process served as a valuable mechanism for refining code quality and enhancing overall project integrity. To facilitate communication, we leveraged Microsoft Teams, utilizing its chat and channel features for seamless interaction and coordination. Additionally, Outlook proved instrumental in managing project-related correspondence. These proficient tools not only streamlined communication but also facilitated efficient collaboration and ensured a productive working environment for the entire team.

C. Project Management

During during BTP, I had the opportunity to delve into the realm of project management. This involved the application of various project management techniques such as meticulous planning, effective resource allocation, task prioritization, and risk assessment. Although my experience in project management was limited, I gained exposure to tools like Jira and Confluence, which facilitated project planning, task tracking, and documentation. Jira, with its issue tracking capabilities, allowed for streamlined project progress monitoring, while Confluence served as a collaborative documentation platform to centralize project - related knowledge. Through these experiences, I gained valuable insights into the importance of project management in ensuring successful project outcomes and the potential benefits of utilizing tools like Jira and Confluence to support project management processes.

D. Learning Methodology Shift and Time Management Skill

The transition from academic learning to industrial learning brings about a notable shift in learning methodology. In academic settings, the emphasis is often on theoretical knowledge acquisition, research-based learning, and individual assessment. On the other hand, industrial learning places a greater emphasis on practical application, hands - on experience, and collaborative problem - solving. During the BTP, I realized that this shift necessitates a reorientation of learning approaches. Industrial learning promotes the development of practical skills, critical thinking, adaptability, and effective teamwork, as individuals engage in project-based tasks and gain exposure to industry - specific tools and practices. As this was my first development - kind project, one thing I experienced and learned throughout the project quite significantly was that the amount of theoretical and in - depth knowledge required for a assigned task has to be given importance. Because otherwise learning a tech stack with multiple components causes many delays. During the initial month of the project, which involved a training period accompanied by hands - on exercises, I was learning almost everything related to a given task in - depth, similar to an academic project, so as a result I was legging so much in learning tech stack for the project. After discussing this my leader, I learned that it's more effective to focus on specific aspects required for a given task and learn other necessary things along the way. This enables better time management and ensures that we acquire the relevant knowledge while progressing through the project.

III. Contributions

A. Migration Summary

- After the creation of the "Private Chats Migration" project and the configuration of its details, encompassing project information, source settings, and destination settings, the project is saved. Upon successful saving, the addition of source and destination users is enabled. Each pair of source and destination users is represented as an individual line item within the project. Subsequently, by selecting the desired line items, the migration of chats can be initiated. Upon selecting a particular line item (representing a specific source and destination user pair), the user will be seamlessly redirected to the corresponding line item detail page.

- As per the stipulated requirement, the aforementioned detail page should encompass a comprehensive summary outlining the total count of chats successfully migrated, along with those that encountered errors. So in the backend, I had to implement an API (controller and its services) which returns migration summary data for the given line item.

- Returned migration summary data from the implemented API comprises the number of successfully migrated 1:1 chats, group chats, and meeting chats, as well as the count of 1:1 chats, group chats, and meeting chats that encountered migration errors.

- As part of its implementation - For the passed project item - ID as a parameter to the API, I fetched all chat threads from azure storage table named "ChatThreads" partitioned by item - ID. Now from these chat threads, I conditionally counted successful and failed 1:1 chats, group chats, and meeting chats. Here to distinguish any of the three kind of chats into successful or failed, I checked its import status or export status. For returning this data I used one of the general Data Transfer Objects (DTO) available in the project.

B. Retry Error

- The requirement for this feature is that sometimes, due to some errors, complete migration cannot take place, resulting in the possibility that some of the chat

threads may not have been migrated in a migration pass. Therefore, to address this issue, there had to be a new feature that allows users to perform migration again, as a subsequent pass, ensuring that all chat threads can be restored. Still there are some errors that cannot be resolved like item-size being too high, corrupted items.

- So here I had to implement an API (controller and its services) in the backend to achieve the aforementioned functionality. As part of implementation, First of all, I needed to check that status and migration type of the project item must be completed and full respectively to perform retry error action. Following to that I needed to get the migration status of selected project items and then return the list of failed project items.

- To check item status, I fetched all chat threads of each of the project item from azure table storage partitioned by item - ID. Here using Azure storage API, I had made a string - formed table query to get chat threads of items and then executed it. And if for at least one of the chat threads, migration were to fail then I had to consider that item as failed. I didn't aware of how to query the azure storage tables through the application code itself, I referred Azure's documentation for the same and learned complex querying.

- To store and return the list of failed project items, I had created a custom Data Transfer Object (DTO) that encapsulates the relevant data, including an array of failed items, their count, and type of migration(here failed). DTOs are frequently employed within N-Tier applications by the Services layer to facilitate seamless data transmission between the Services layer and the UI layer [3].

C. RnD - Team Chat Data Storage

- I was assigned the task of researching and documenting the storage mechanism employed for Teams chat data. This involved a thorough examination of the codebase of previous projects, leading me to discover that Teams chat data is predominantly stored using cloud storage services such as Azure Blob and Amazon S3 bucket, adhering to the specific requirements of each project. To facilitate seamless access to these cloud storage resources, pertinent information pertaining to their corresponding access credentials is diligently stored within the SQL tables.

- During my exploration, I delved into the intricacies of Azure Table Storage Design Patterns, aiming to enhance the performance and efficiency of the current project. By harnessing the power of these design patterns, I aspired to optimize the retrieval and manipulation of Teams chat data, ensuring swift and streamlined operations. This meticulous investigation and implementation of Azure Table Storage Design Patterns were instrumental in fortifying the overall performance and effectiveness of the project, enabling smooth interactions with the stored chat data.

D. Resolving Select All Functionality

- The detail page for every project contained rows for each line item added in that project. There was a button named 'select all' in the menu bar of platform, on clicking which selects all the line items in the project. There are several functionalities, i.e. migration, clean-up, retry error etc, available to perform on selected line items in the project. After selecting some line items manually, and then if I were to perform any of these functionalities then it worked properly, but any of the functionalities with selecting all the line items through that 'select all' button was not working. I was tasked with investigating and resolving this critical issue that arose within the project. The challenge at hand required a comprehensive analysis to identify the root cause of the problem and devise an effective solution.

- I tested all of the above functionalities with both working and non - working scenarios, that's manually selecting individual line items and using the 'select all' button. I found that in the former case, the payload contained the IDs of all the selected line items, while in the latter case, the payload didn't contain any IDs (it was empty). So made changes in (FE or BE?) that whenever 'select all' is used, IDs of all line items must passed in payload like in manual selection case. Here I explored the powerful API development and testing tool, Postman.

E. Setup and Configuration of Development Environment for Project Execution

- During the course of the project, a significant contribution I made was setting up the complex development environment required for running the project on the local machines of the testing teams. This involved the installation and configuration of various essential tools and technologies. Firstly, I ensured the setup of Java, NodeJs, and EmberJs for the front-end development, enabling the team to develop and test the client-side applications effectively.

- Additionally, I facilitated the installation of C# and related frameworks, such as Visual Studio and Azure Functions, to support the development of the back-end components. The integration of these technologies provided a robust foundation for the project's functionality. Furthermore, I guided the testing teams in installing relevant tools such as Visual Studio Code for code review and collaboration. By meticulously setting up the development environment, I aimed to streamline the testing process and enhance the team's productivity.

IV. Conclusion

Throughout the duration of my BTP, I had the invaluable opportunity to work on live projects, immersing myself in the world of software development industry standards and best practices. This hands-on experience allowed me to grasp the workings of agile development methodologies, emphasizing the significance of adhering to coding standards and adopting efficient tools and technologies. Regular team meetings and client interactions further underscored the importance of clear communication and constructive feedback in the product development process. In addition to acquiring these valuable skills, I successfully completed all assigned tasks, contributing to the development of essential functionality for MigrationWiz.

Acknowledgement

Engaging in this BTP has proven to be an invaluable educational journey, allowing me to acquire a plethora of new skills and expand my knowledge significantly. I appreciate Prof. Abhishek Jindal. for keeping an eye on us throughout the BTP project. I extend my heartfelt gratitude to my off - campus mentor, Deep Dave, for affording me the opportunity to contribute to and guiding me in this live project, which will undoubtedly prove beneficial for my future endeavors.

References

[1] https://www.bittitan.com/migrationwiz/why-migrationwiz
[2] https://medium.com/aeturnuminc/what-is-ember-js-ff94403fec96
[3] https://stackoverflow.com/questions/1051182/what-is-a-data-transfer-object-dto

201901306_Final_Submission_Plag

ORIGINALITY REPORT

4%	1%	0%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Dhirubhai Ambani Institute of Information and Communication Student Paper	4%
2	Submitted to University of East London Student Paper	1%
3	www.microsoft.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		