

INTERNSHIP REPORT

Exponential AI

Sreeshanth S | 21WU0101017 | AI & DS

Submitted to:



Woxsen University

Dean: Dr. Pep Lluís Esteva

Program Director: Prof. Balaji Venkateswaran

Point of Contact: Prof. Bhanu Prakash

Company:



Exponential AI

Point of Contact: Gaurav Agarwal

Declaration

I hereby declare that this internship report is an authentic representation of my own original work and reflections during my internship tenure. This report contains my experiences, insights, and contributions gained through active engagement during my internship at Exponential AI. I acknowledge that any form of academic dishonesty, including plagiarism or fabrication, undermines the integrity of this report and my academic pursuits. Therefore, I affirm my commitment to upholding the principles of academic integrity and honesty throughout the preparation and submission of this internship report.

Sreeshanth S

21WU0101017

27/11/2024

Acknowledgement

I would like to thank Mr. Gaurav Agarwal and the entire team at Exponential AI for providing me with a conducive learning environment and valuable opportunities to apply my skills and knowledge in real-world scenarios.

I extend my sincere appreciation to Dr. Pep Lluís Esteva, Dean of the School of Technology (SOT), Prof. Balaji Venkateswaran, Program Director, and Prof. Bhanu Prakash for their unwavering support and encouragement of students seeking industry exposure and practical experience.

I would also like to express my gratitude to my colleagues at Exponential AI for their ongoing support, commitment to maintaining a positive work culture, and for contributing to a rewarding experience.

Abstract

This internship report provides an overview of my ongoing experiences and learning journey at Exponential AI. It highlights the tasks and responsibilities I have undertaken as part of the UI Automation team, with a primary focus on automating functionalities for an Intelligent Document Processing platform. My work has centered around Behavior Driven Development (BDD) using Cucumber feature files and Selenium in Python to ensure the platform's functionalities are comprehensively tested and robust.

I have gained invaluable insights into the product by exploring its workflows and understanding the requirements for automation testing. Additionally, I have honed my skills in test script development, debugging, and integrating automation frameworks into a CI/CD pipeline using Docker.

My efforts have been dedicated to ensuring all platform functionalities are thoroughly automated, enabling the seamless identification of bugs and regression issues. This has been achieved through collaboration with my team and iterative testing of key modules such as document processing, OCR extraction, and ML model workflows.

Currently, I am focused on refining existing automation scripts and contributing to the development of a pipeline that runs all automated tests, ensuring the platform's reliability for future releases. This internship experience has been instrumental in expanding my technical expertise in UI Automation and software quality assurance.

Introduction

Artificial Intelligence (AI) and Machine Learning (ML) are transforming industries by revolutionizing how businesses operate, make decisions, and innovate. In the healthcare sector, AI and ML have the potential to reshape patient care, diagnosis, and treatment by enabling data-driven decision-making. By analyzing large volumes of medical data, these technologies facilitate early disease detection, personalized treatment plans, and precise diagnostic imaging, paving the way for improved outcomes and efficiency.

In this context, **Exponential AI** offers **Enso**, a cutting-edge platform that empowers users to harness the power of AI-driven decision-making. Enso enables users to design, train, and manage decision agents—digital employees equipped with human-like decision-making capabilities. These decision agents are highly adaptable and trainable, allowing users to customize them for specific datasets, domains, or market conditions. This flexibility ensures optimal results across a variety of use cases, especially in complex and dynamic fields like healthcare.

During my internship at Exponential AI, I have worked extensively as part of the **UI Automation** team, focusing on automating the platform's user interface functionalities. My work has involved leveraging tools such as Selenium in Python and Behavior Driven Development (BDD) using Cucumber to create robust automation frameworks. This effort aims to streamline the testing process for the Enso platform, ensuring reliability and efficiency in detecting bugs and verifying feature functionality.

By contributing to the automation of core functionalities such as document processing, OCR workflows, and ML model operations, I have gained valuable experience in designing and debugging automation scripts. Additionally, I am working closely with my team

to integrate these automated scripts into a CI/CD pipeline using Docker, enabling seamless execution and validation of tests across multiple platform modules. This hands-on experience has significantly expanded my expertise in UI Automation, quality assurance, and collaborative software development.

Internship Experience

1. Roles and Responsibilities

- **Development and Maintenance of Automated Test Scripts:** Designed and implemented comprehensive test scripts to automate various functionalities of the Enso platform. This included critical workflows like document processing, OCR text extraction, and machine learning model operations. The scripts were crafted to ensure robust coverage of platform features and adaptability for future updates.
- **Behavior Driven Development (BDD) Implementation:** Employed Cucumber to define user behavior scenarios through feature files. Each scenario described specific interactions on the platform, and step implementations were coded in Python to automate these behaviors. This approach ensured a clear and collaborative understanding of requirements between developers and testers.
- **Collaborative Test Scenario Identification:** Participated in team discussions to prioritize functionalities for automation. This involved analyzing the platform's workflows, identifying high-risk areas prone to bugs, and ensuring test scenarios aligned with real-world use cases.
- **Debugging and Quality Assurance:** Actively debugged and resolved issues in test scripts, ensuring stability and reliability. Quality checks were conducted to ensure scripts adhered to coding standards and seamlessly integrated into the overall automation framework.
- **CI/CD Pipeline Preparation:** Currently contributing to developing a Dockerized pipeline for executing all automated

tests in an integrated environment. This ensured efficient and consistent validation of platform functionalities across different modules.

2. Learning Outcomes

- **Expertise in Selenium for UI Automation:** Gained hands-on experience with Selenium, mastering its core capabilities, including handling dynamic elements, explicit waits, and browser interactions.
- **Proficiency in Behavior Driven Development (BDD):** Acquired a deep understanding of how BDD fosters collaboration by bridging the gap between technical and non-technical stakeholders. Learned to design clear and structured feature files that accurately reflect user requirements.
- **Insights into CI/CD Integration:** Understood the importance of incorporating automation into continuous integration and delivery pipelines. Learned to work with Docker to containerize tests and optimize pipeline execution.
- **Team Collaboration and Agile Practices:** Improved my ability to collaborate within a team, share knowledge, and work in sprints to achieve set goals. The experience enhanced my understanding of agile development and iterative improvements.

Project Overview

The objective of my work as part of the UI Automation team at Exponential AI was to automate the functionalities of the **Enso** platform, an Intelligent Document Processing platform designed to assist users in creating, managing, and deploying AI-driven decision agents. The automation aims to ensure that platform functionalities are thoroughly tested, enabling seamless detection of bugs in new and existing features, while significantly reducing the manual effort involved in quality assurance.

The automation framework I contributed to focuses on streamlining the testing process for several core modules, such as:

1. Document Processing Workflow

This workflow allows users to upload and process documents using customizable rules. Key steps in the process include:

- **Document Uploading:** Users upload documents to the platform. The system ensures document compatibility and queues the files for processing.
- **OCR Processing:** The platform leverages Google Tesseract to extract text from documents. This involves noise reduction and preprocessing steps to improve text extraction accuracy.
- **Rule Definition and Application:** Users can define rules for document processing, such as extracting specific fields or data formats. These rules are applied during processing to generate structured outputs that align with user requirements.

My work automated the testing of these workflows, ensuring that rules are applied correctly, text is extracted as expected, and outputs match user-defined parameters.

2. Machine Learning Model Workflows

The platform allows users to train, test, and deploy machine learning models for diverse applications. The key steps include:

- **Dataset Management:** Users upload and organize datasets, which are used for training and testing models.
- **Model Training:** Models are trained on user-specified data, using hyperparameters and configurations set by the user.
- **Model Deployment:** Trained models are deployed to handle real-time data processing. Automation scripts were developed to test the entire model lifecycle, including dataset uploading, training processes, and verifying deployed model outputs for accuracy and performance.

3. UI Functionalities and User Interactions

Given the user-centric nature of the platform, testing the UI components and user interactions was essential. This included:

- Navigation through various platform modules.
- Verifying the responsiveness and reliability of key UI elements such as forms, buttons, and dashboards.
- Ensuring that features like notifications and error messages displayed appropriately during operations.

4. Integration with CI/CD Pipelines

To enhance the efficiency of testing workflows, the automation scripts are being integrated into a Dockerized CI/CD pipeline. This ensures that:

- All automated tests can be executed seamlessly in a consistent, integrated environment.

- Bugs and regressions are identified early during the development cycle, reducing the risk of issues in production releases.

The automation framework is critical to achieving the platform's long-term goal of full functionality testing, ensuring reliability and robustness for future releases and updates. This project has allowed me to gain a deep understanding of automation testing, platform workflows, and collaborative software development practices.

Methodology

The UI Automation efforts for the **Enso** platform followed a structured approach that ensured comprehensive coverage of all critical functionalities. The process included a clear understanding of workflows, developing reusable automation scripts, and integrating them into a robust testing pipeline. Below is a detailed overview:

1. Requirement Analysis

The first step involved thoroughly analyzing the platform's key workflows to identify areas suitable for automation.

- Collaborated with the development team to understand the end-to-end **document processing workflow**, including document upload, preprocessing, rule application, and text extraction using Google Tesseract OCR. Special focus was given to validating custom rules defined by users and ensuring extracted text matched these rules.
- Studied **machine learning model workflows** to automate test cases related to dataset uploads, model training, and deployment. The analysis focused on critical checkpoints, such as ensuring the integrity of uploaded datasets and verifying the accuracy of model predictions post-deployment.
- Prioritized high-impact functionalities and workflows to create an efficient testing strategy, ensuring major use cases and edge scenarios were accounted for in test scripts.

2. Tool Setup and Environment

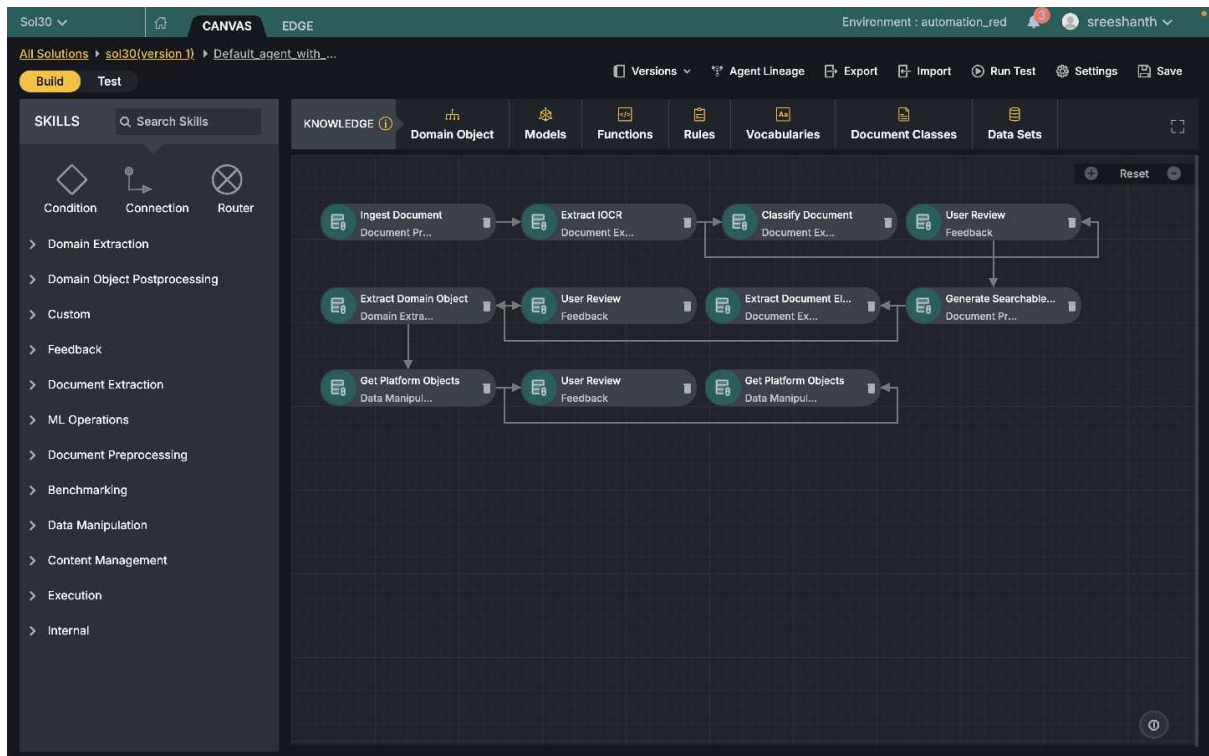
A robust testing framework was established using modern tools and methodologies:

- **Selenium** was used to interact with the browser, automating complex UI components like file uploads, dashboards, and navigation through various modules.
- Adopted **Behavior Driven Development (BDD)** with Cucumber to create human-readable feature files. These files described user scenarios, such as validating successful document uploads and rule-based processing, and were implemented using Python.
- Test environments were configured for cross-browser testing to validate platform reliability across different user setups, ensuring consistent behavior for all workflows, including document and model management.

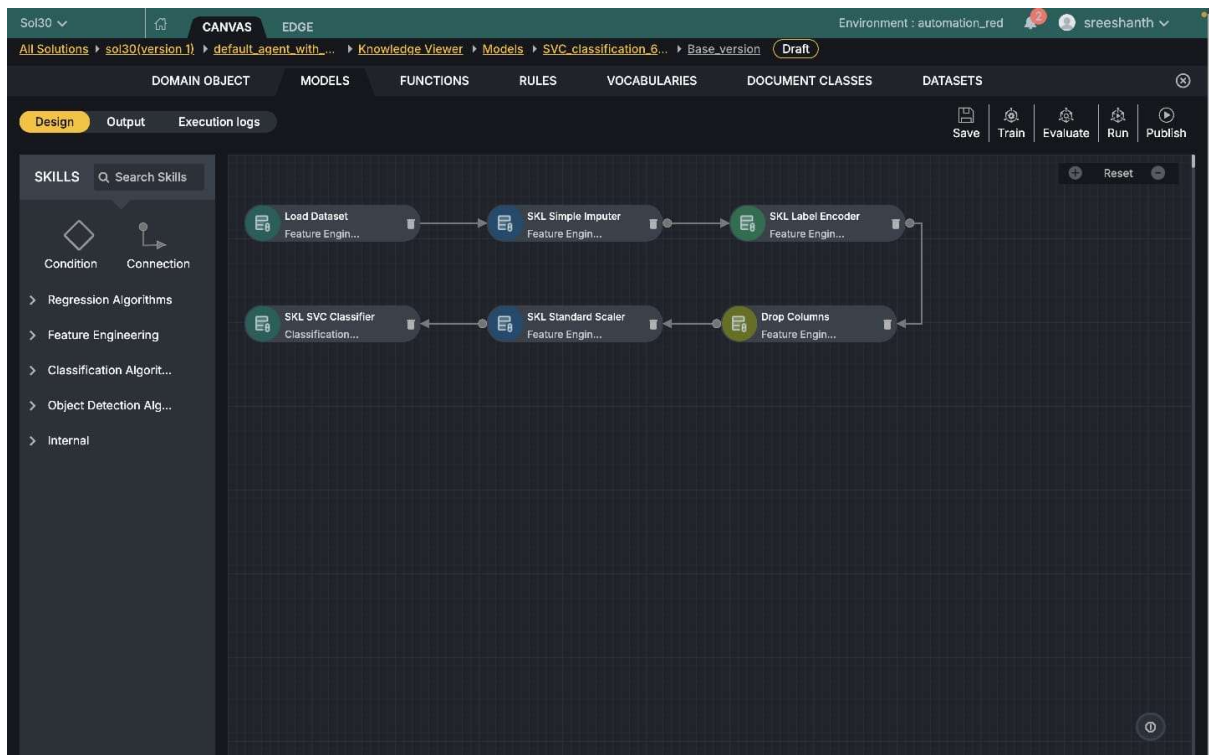
3. Test Case Development

Detailed test cases were designed to cover critical workflows while maintaining modularity and reusability:

- Automated the **document processing workflow**, verifying that uploaded documents were processed as expected, custom rules were applied accurately, and text extracted via OCR was free of errors. Tests also handled edge cases, such as unsupported or corrupted document inputs.



- For **model workflows**, test scripts ensured datasets were uploaded and organized correctly, models were trained on the right configurations, and deployed models performed accurate predictions. End-to-end validations were conducted to simulate real-world usage scenarios.



- Automated UI interactions to validate core platform functionalities, including form submissions, navigation, and responsiveness of dashboards.
- Each script was designed to handle error scenarios gracefully, ensuring that bugs or regressions could be identified effectively.

4. Pipeline Integration

Currently contributing to the development of a Dockerized CI/CD pipeline to integrate and streamline the automated testing process:

- All automation scripts are containerized to maintain consistency across different environments.
- The pipeline enables running all test scenarios in an integrated manner, including document processing and model workflows, ensuring seamless validation of platform functionalities during every build.
- By reducing manual testing effort, the pipeline accelerates bug detection and provides faster feedback for feature enhancements or fixes, contributing to the platform's robustness and reliability.

This structured methodology ensures that automation scripts effectively validate platform workflows, including document processing and machine learning functionalities, while the integration into a CI/CD pipeline prepares the platform for efficient, future-proof quality assurance processes.

Results and Insights

The UI automation efforts for the **Enso** platform have yielded significant improvements in both testing efficiency and platform stability:

1. Automation Achievements

- Key workflows, including **document processing** (OCR validation, rule application) and **model workflows** (dataset uploads, model training, and deployment), have been successfully automated.
- **UI Features** such as form validations, dashboard rendering, and navigation have also been validated through automation, ensuring comprehensive test coverage across the platform.

2. Enhanced Testing Efficiency

- Automation reduced testing time, enabling faster validation of platform functionalities. Tasks that once took hours are now completed in minutes.
- The use of reusable and modular scripts ensures scalability and consistency in test execution.

3. Bug Identification and Resolution

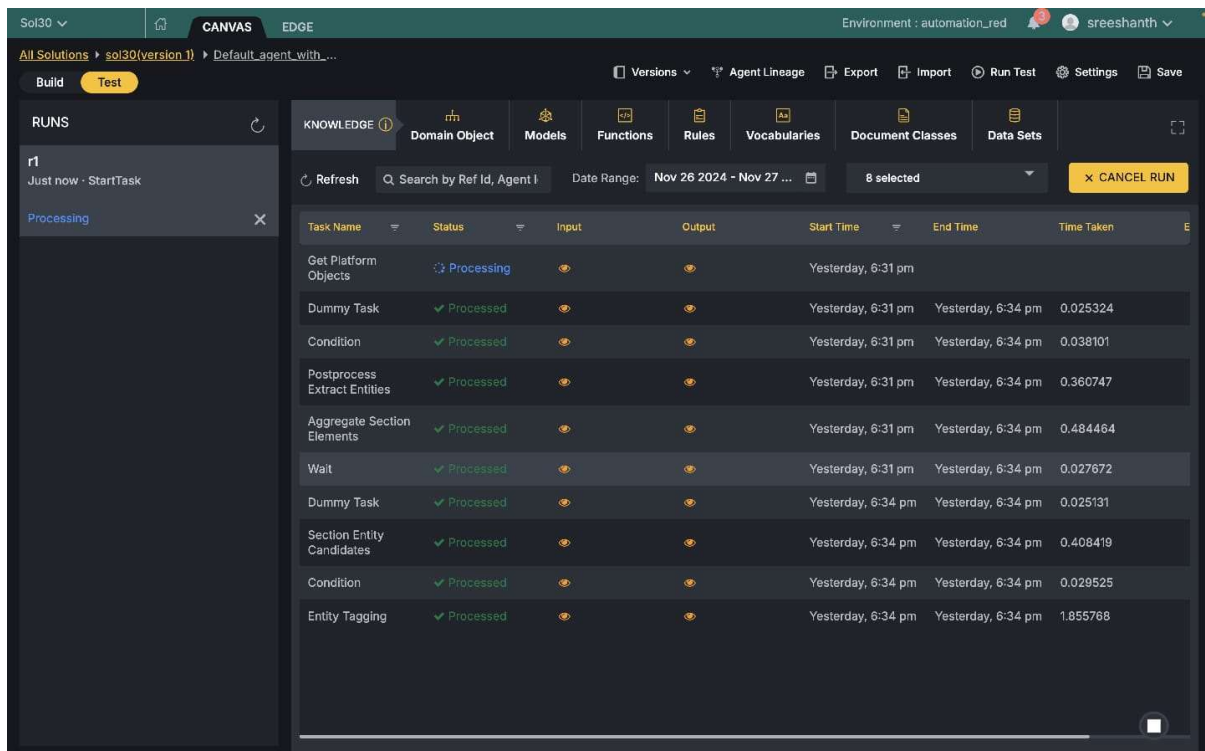
- Automation identified bugs in **OCR accuracy**, **model deployment workflows**, and **UI inconsistencies**. These bugs were promptly fixed, improving platform stability and reliability.

4. Preparation for CI/CD Integration

- The ongoing development of a **Dockerized CI/CD pipeline** is expected to further streamline testing, ensuring consistent validation of all workflows during every build.
- This pipeline will speed up bug detection and facilitate faster feature rollouts.

Key Insights

- Automation has significantly enhanced testing efficiency, providing deeper validation coverage.
- The integration of automated tests into the pipeline is crucial for long-term testing scalability, reducing manual effort and accelerating feedback for future platform releases.



The screenshot displays the Sol30 application interface. The top navigation bar includes 'Sol30', 'CANVAS', and 'EDGE'. The environment is set to 'automation_red'. The main interface shows a 'RUNS' panel on the left with a 'Processing' status. The central panel displays a table of task execution results. The table has columns for Task Name, Status, Input, Output, Start Time, End Time, and Time Taken. The tasks listed are: Get Platform Objects (Processing), Dummy Task (Processed), Condition (Processed), Postprocess Extract Entities (Processed), Aggregate Section Elements (Processed), Wait (Processed), Dummy Task (Processed), Section Entity Candidates (Processed), Condition (Processed), and Entity Tagging (Processed). The 'Time Taken' column shows values ranging from 0.025324 to 1.855768.

Task Name	Status	Input	Output	Start Time	End Time	Time Taken
Get Platform Objects	Processing			Yesterday, 6:31 pm		
Dummy Task	Processed			Yesterday, 6:31 pm	Yesterday, 6:34 pm	0.025324
Condition	Processed			Yesterday, 6:31 pm	Yesterday, 6:34 pm	0.038101
Postprocess Extract Entities	Processed			Yesterday, 6:31 pm	Yesterday, 6:34 pm	0.360747
Aggregate Section Elements	Processed			Yesterday, 6:31 pm	Yesterday, 6:34 pm	0.484464
Wait	Processed			Yesterday, 6:31 pm	Yesterday, 6:34 pm	0.027672
Dummy Task	Processed			Yesterday, 6:34 pm	Yesterday, 6:34 pm	0.025131
Section Entity Candidates	Processed			Yesterday, 6:34 pm	Yesterday, 6:34 pm	0.408419
Condition	Processed			Yesterday, 6:34 pm	Yesterday, 6:34 pm	0.029525
Entity Tagging	Processed			Yesterday, 6:34 pm	Yesterday, 6:34 pm	1.855768

Challenges and Conclusions

1. Challenge: Difficulty in replicating user scenarios.

Solution: Since the platform involves complex user interactions, replicating real-world user scenarios for automated testing was challenging. To address this, I collaborated closely with the product team to better understand user workflows and expectations. By designing more comprehensive and detailed feature files, we could simulate real user behavior more accurately. This approach ensured that the tests captured a wider range of scenarios and led to more realistic validations of platform functionalities.

2. Challenge: Managing script failures due to dynamic UI elements.

Solution: Dynamic UI elements, such as frequently changing buttons, dropdowns, and input fields, often caused automation scripts to fail. To solve this, I implemented more robust locators using unique attributes like CSS selectors or XPath, which are less prone to changes. Additionally, I incorporated Selenium's explicit waits to handle synchronization issues, ensuring the script waits for elements to load or become interactable before performing actions. This improved the stability and reliability of the tests, reducing the number of failures due to dynamic UI changes.

3. Challenge: Pipeline integration complexities.

Solution: Integrating the automation scripts into a Dockerized CI/CD pipeline presented several challenges, including environment setup issues and compatibility with different modules. To overcome this, I took an iterative approach, continuously debugging the setup and configuring the environment to ensure compatibility. By testing the pipeline incrementally, I was able to isolate and resolve issues, leading to the successful execution of automated tests within the CI/CD framework. This ensured that every code update triggered automatic validation of platform functionalities, streamlining the testing process.

Conclusion

My internship at Exponential AI provided an excellent opportunity to dive into the world of UI automation within the Enso platform.

Initially, I focused on understanding the platform's core functionalities and identifying key workflows that required automated testing. This deep dive into the platform, coupled with my collaboration with the product team, allowed me to identify areas where automation could be implemented to streamline testing processes and ensure platform stability. I gained valuable experience with tools like Selenium in Python and behavior-driven development (BDD) using Cucumber feature files, which helped in designing clear, concise, and effective test cases.

As my role evolved, I began working on automating critical platform features, such as document processing, model workflows, and user interface validations. By implementing reusable and modular test scripts, I was able to ensure comprehensive coverage of platform functionalities. Additionally, I focused on managing challenges such as dynamic UI elements and replicating real user scenarios, improving the overall efficiency and stability of the automated tests. One of the significant milestones in my internship has been contributing to the development of a Dockerized CI/CD pipeline, which will ultimately enable the seamless execution of all automated tests across different modules, ensuring consistent quality checks and faster bug identification for future releases.

Through this experience, I have gained a deeper understanding of the importance of automated testing in ensuring software reliability and have honed my skills in developing scalable, maintainable test frameworks. By continuing to improve and expand the automation efforts, I am confident that my contributions will significantly enhance the quality assurance processes at Exponential AI, ensuring that the Enso platform remains stable and robust as it continues to evolve.

Annexure

Offer Letter:



Internship Letter

Date: 11th March 2024
Name: Sreeshanth S

Dear Sreeshanth S,

We are pleased to confirm your acceptance of an internship position as “**Python Intern**” in the **Platform Division** at a pay rate of **25,000 INR** per Month. The duration of your internship is for **three months** on probation i.e., from **13th March 2024** to **30th June 2024**. Your duties and assignments for this position will be those described to you in your orientation with **Mr. Gaurav Agarwal**.

If you have any questions, please feel free to contact Gaurav or me. We are very pleased that you have decided to join **Exponential AI**. We look forward to seeing you on **13th March 2024** and offer a very warm welcome.

Please give us your written acceptance of the above-mentioned terms by returning to us the duplicate copy of this letter duly signed by you and also indicating the date of joining.

A hearty welcome.

Sincerely,
For Exponential AI

A handwritten signature in blue ink, appearing to read "Mukesh".

Mukesh Kumar Murugan
Director - HR

A handwritten signature in blue ink, appearing to read "Sreeshanth".

Accepted
Sreeshanth S

Extension Letter:



Internship Letter

Date: 27th June 2024

Name: Sreeshanth Reddy S

Dear Sreeshanth,

I hope this email finds you well. We are pleased to inform you that we have extended your internship until December 31, 2024.

We appreciate your contributions to the team and look forward to your continued growth and development with us.

Please confirm your acceptance of this extension by replying to this email. If you have any questions, feel free to reach out to HR Team.

Sincerely,
For Exponential AI

A handwritten signature in blue ink, appearing to read "Mukesh".

Mukesh Kumar Murugan
Director - HR

Accepted
Sreeshanth Reddy S

Exponential AI CIN: U72900TG2016PTC109114

Reg. Office: Unit no: 1005, 10th Floor, Jain Sadguru Image Capital Park, Image Gardens Rd, VIP Hills, Jaihind Enclave,
Madhapur, Hyderabad, Telangana, 500081