Turnitin Originality Report

Processed on: 08-Dec-2023 13:56 PKT
ID: 2250985006
Word Count: 11187
Submitted: 2

Introduction By Hadi Shehryar

Similarity Index	Similarity by Source	
13%	Internet Sources:	4%
	Publications:	0%
	Student Papers:	12%

3% match (student papers from 18-Dec-2014)
Submitted to Higher Education Commission Pakistan on 2014-12-18

2% match (student papers from 02-Dec-2022)
Submitted to University of Hertfordshire on 2022-12-02

1% match (student papers from 15-Dec-2012)
Submitted to Higher Education Commission Pakistan on 2012-12-15

1% match (student papers from 10-Nov-2012)
Submitted to Higher Education Commission Pakistan on 2012-11-10

1% match (student papers from 03-Jun-2020)
Submitted to Higher Education Commission Pakistan on 2020-06-03

1% match (student papers from 08-Jun-2023)
Submitted to Higher Education Commission Pakistan on 2023-06-08

1% match (student papers from 03-Dec-2019)
Submitted to Higher Education Commission Pakistan on 2019-12-03

1% match (student papers from 16-May-2016)
Submitted to Arab Open University on 2016-05-16

1% match (Internet from 15-May-2023)
https://www.coursehero.com/file/p6hvi12o/Table-364-Request-Contact-Number-This-table-describes-the-use-case-to-request/

< 1% match (student papers from 24-Nov-2023)

Submitted to Higher Education Commission Pakistan on 2023-11-24

< 1% match (student papers from 06-Jun-2014)
Submitted to Higher Education Commission Pakistan on 2014-06-06

< 1% match (student papers from 20-Mar-2018)
Submitted to Higher Education Commission Pakistan on 2018-03-20

< 1% match (student papers from 23-Sep-2011)
Submitted to Higher Education Commission Pakistan on 2011-09-23

< 1% match (student papers from 09-May-2013)
Submitted to Higher Education Commission Pakistan on 2013-05-09

< 1% match (student papers from 07-May-2019)
Submitted to Higher Education Commission Pakistan on 2019-05-07

< 1% match (student papers from 10-Dec-2013)
Submitted to Higher Education Commission Pakistan on 2013-12-10

< 1% match (student papers from 16-May-2016)
Submitted to Arab Open University on 2016-05-16

< 1% match (student papers from 09-Jul-2020)
Submitted to National College of Ireland on 2020-07-09

< 1% match (student papers from 16-Jun-2019)
Submitted to National College of Ireland on 2019-06-16

< 1% match (Internet from 29-Apr-2021)
http://docplayer.net/8944983-Real-time-embedded-panoramic-imaging-for-spherical-camera-system.html

< 1% match (Internet from 25-Jan-2017)
http://docplayer.net/13917034-Sdp-element-concept-science-data-processor.html

< 1% match (student papers from 26-Jul-2023)
Submitted to Technological University Dublin on 2023-07-26

< 1% match (student papers from 26-Nov-2023)
Submitted to University of Alabama at Birmingham on 2023-11-26

< 1% match (publications)

康希 脇田, 敦生 牧, 洋平 秋本, 凛 巣山, 新喜 宮内, 佑介 青木, 直哉 梅田. "2021A-OS1-3 着桟制御に向けた不規則風外乱下における経路追従及び定点保持制御の強化学習を用いた獲得手法について", 日本船舶海洋工学会講演会論文集, 2021

< 1% match (student papers from 14-Oct-2021)
Submitted to Chester College of Higher Education on 2021-10-14

< 1% match (student papers from 01-Dec-2023)
Submitted to University of Strathclyde on 2023-12-01

< 1% match (student papers from 02-Jul-2022)
Submitted to Xiamen University on 2022-07-02

< 1% match (student papers from 15-Oct-2023)
Submitted to Delaware Technical Community Wilmington on 2023-10-15

< 1% match (student papers from 03-Dec-2012)
Submitted to Colorado Technical University Online on 2012-12-03

< 1% match (student papers from 15-Aug-2019)
Submitted to Universiti Tunku Abdul Rahman on 2019-08-15

< 1% match (Internet from 19-Dec-2022)
https://edoc.pub/virtual-stock-exchange-pdf-free.html

< 1% match (student papers from 09-Apr-2015)
Submitted to Glasgow Caledonian University on 2015-04-09

< 1% match (student papers from 22-Jan-2013)
Submitted to Binary University College on 2013-01-22

< 1% match (Internet from 12-Oct-2023)
https://netleon.com/blog/full-stack-web-development-company-india/

< 1% match (Qamar Askari, Irfan Younas, Mehreen Saeed. "Emphasizing the importance of shift invariance in metaheuristics by using whale optimization algorithm as a test bed", Soft Computing, 2021)
Qamar Askari, Irfan Younas, Mehreen Saeed. "Emphasizing the importance of shift invariance in metaheuristics by using whale optimization algorithm as a test bed", Soft Computing, 2021

National University of Computer and Emerging Sciences Scholar Support FYP Team Ibrahim Maaz.....19L-1109 Usman Elahi.....19L-1223 Malik Hadi Shehryar.....19L-0955 Supervised by M. Naveed FAST

School of Computing National University of Computer and Emerging Sciences Lahore, Pakistan December 2023 Anti-Plagiarism Declaration This is to declare that the above publication produced under the: Title: ____Scholar Support_____ is the sole contribution of the author(s) and no part hereof has been reproduced on as it is basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined. Date: March 10, 2023 Student 1 Name: Ibrahim Maaz Signature: Student 2 Name: Usman Elahi Signature: Student 3 Name: Malik Hadi Shehryar Umar Signature:

_____ Authors' Declaration This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization. Abstract The Scholar Support Platform offers a solution for students who need help paying for their tuition fees. It connects students with donors who can provide support through loans or donations. This platform makes the process of seeking and receiving assistance safe, easy and efficient. With its user interface secure payment processing and open communication channels it creates a way for students to achieve their academic goals. Additionally, it fosters a community of individuals who are committed to supporting learners and ensuring their success. The platform's goals and objectives, as well as its scope, aim to provide a comprehensive solution that addresses the financial challenges faced by students and empowers them to take control of their financial future. Executive Summary Scholar support connects donors with students. It is clear that a better solution to our problem is required critically and that would fill the gap in Pakistan's educational industry. Current organizations do not provide the power to choose who they want to provide money but with Scholar Support they can find a student whom they wish to help and in addition to that out integrated recommendation system would help them in that cause. Scholar Support 1 Introduction The Scholar Support Platform project intends to build a platform where students may request contributions or loans from people who have joined up to help them pay for their education. The platform will provide an easy, safe, and dependable means for students to obtain financial assistance and donors to give to a worthwhile cause. This report explains the project's major objectives, scope, and limits, as well as the stakeholders' high-level aims and challenges. The Scholar Support Platform project aims to provide a new avenue for students to obtain financial assistance for their education. With rising tuition costs and limited access to funding, many students struggle to pay for their studies. This platform aims to bridge the gap between students in need of financial support and donors who are willing to contribute to a worthy cause. The platform will provide a safe, easy, and reliable way for students to request contributions or loans from donors who have signed up to support their education. The project's main objective is to create a user-friendly platform that streamlines the donation process and makes it easy for students to connect with potential donors. This report will outline the project's scope and limits, including the stakeholders' high-level objectives and challenges. The project team recognizes the potential challenges and limitations of the platform, including ensuring the safety and security of financial transactions and maintaining the privacy of both students and donors. Overall, the Scholar Support Platform project is a noble effort to address the financial challenges facing many students today. By creating a platform that connects students with donors, the project hopes to improve access to education for all and contribute to a more equitable society. The first section gives an introduction of the project. The second section gives an overview of the project, it discusses the scope and goal of the project. The third section gives a brief overview of the related work. The fourth section of the document gives an overview on how the project would be implemented. It has some portion of the SRS document 1.1 Purpose of this

Document This document provides an overview of the Scholar Support system. Its goals, objectives, members and audience. Describes the project and the possible impact it has. 1.2 Intended Audience They are investors, stakeholders, project team members, students, and those eager to support the platform's goal of helping students with financial aid. 1.3 Definitions, Acronyms, and Abbreviations List all important definitions, the acronyms and abbreviations used in this document. IDE: Integrated Development Environment API: Application Programming Interface DOS: Denial of Service Literature Review / Related Work This chapter will put a focus on organizations and projects that are already existing and have same ideology or problem statement as Scholar Support. 2.1 Detailed Literature Review In this section we will review the organizations that are working for student funds or donations. The strengths and weaknesses of these organizations will also be put under the microscope in this section. 2.1.1 GoFundMe Summary of the research item Currently GoFundMe is a massive platform that allows individuals to obtain finances for various needs which includes educational needs as well. It is Web based so it connects people globally due to its large active userbase and according to the giving report of 2021, the GoFundMe community has raised over 15 billion Dollars for cases [1]. A person posts a fundraiser on the platform and people donate according to their will and budget. GoFundMe platform also provides the option for donors to send supportive messages for emotional support. The platform also charges a transaction fee of 2.9%+$0.30 which is charged from the donor. Any person can set up account for himself or on the behalf of the other person, he will tell his story and set a goal which is the total amount he expects to receive. If a person does not wish to use his name, he can post the fundraiser anonymously as well. Critical analysis of the research item It has both strengths and weaknesses. If we analyze what made this platform so successful, we look at its strengths. Firstly, is its reliable payment method in which the user has a peace of mind that after doing adequate research on the recipient, he knows that after he has donated the amount will reach the receiver, moreover the GoFundMe team also verifies documents, bank statement of the beneficiary. Potential fraudsters can be reported to the Safety and Trust team. Donors may also receive a refund if the organizer or beneficiary of the fundraiser commits a misuse. It has earned its reputation over time for being a dependable platform for fundraising where donations delivered securely to the right person. GoFundMe also is strongly integrated with social media apps such as Instagram and Facebook. A fundraiser Instagram story can be uploaded each week with an update on the fundraiser. A fundraiser link can be provided on social media so your friends and family can see as well and spread the word and in this way your reach is increased. It has an easy-to-use interface which provides all the functionality and doesn’t have a steep learning curve. It has weaknesses as well as the fundraising campaigns have a strong focus on emotional appeals, for example the more distressing the story is like people affected by a disaster are more likely to get funds, educational cases are less likely to. Similarly, there is inequality in some cases where cases more significance are given less importance only because they didn’t have an interesting story or digital tools were not properly used to portray the story [2]. Even after all the strict measures, fraudsters will find ways to exploit it. Occasionally fake accounts are created which are misused and people are scammed. Relationship to the proposed research work The relationship of our proposed project with GoFundMe is that we also intend to make a platform where donors will donate to beneficiaries but instead of broadening and diverting our scope we would focus on educational aid and make it better. Our target audience would a part of the target audience of GoFundMe. We would also ensure the highest possible security and data protocols that GoFundMe adopts as well and have a similar anonymity option. Our payments would be secure. 2.1.2 Akhuwat Foundation Summary of the research item Akhuwat foundation provides interest free loans to people in need. It provides numerous loans which include educational loans. It has a sophisticated system

where the person who applies is thoroughly verified which includes numerous visits of case officers to the borrower's neighborhood, a meeting at his house, a business analysis is performed and finally the loan is recommended or rejected. Even after the loan is given monitoring is done that it is used as stated for example if the purpose of the loan was for education, they follow up that the loan was used for education. Critical analysis of the research item A very professional system is in place which ensures that the loans are used for the right purpose. Has a large donor network around the world so it has vast financial resources resultantly loans are interest free. They also have an experinced team which has dealt with many cases. Since they also have to focus on other loans such as poverty alleviation and business venture related ones, there is no guarantee that enough educational loans are provided. Neither is there any quota system for educational loans. There is an upper limit of loan which is 70,000 rupees and cannot be increased so a person whose fees is more than this amount will also have to arrange the rest [3] They also have a strict criterion which must be met then only the loan is approved, such as such as parents' income is in a certain threshold, and only can be applied once. Since the board of Akhuwat Foundation decides which person will be given the loan, the actual owner of donation money would not be able to know to whom his money went to, and the donor would have to put his entire faith on the foundation to use funds correctly. His money could be given to a housing related case even if his intention was to spend it on an education case. Relationship to the proposed research work We will also provide loans but donations as well. We have left this option to our users whether they want to give loans or donations. Akhuwat foundation offers finances for numerous causes but we have limited ourselves to education so we could focus on it and do so in the best possible manner. Our projects demographic and operational procedure is like Akhuwat Foundation, but we tend to deal with educational cases only and adding more functionality and improving upon it. 2.1.3 National Bank Student Loan Scheme Summary of the research item The National Bank of Pakistan provides a scheme called the National Bank Student Loan Program to assist students in paying for their higher education. This program was initiated by the Government. Eligible students from eligible universities can apply all over from all over Pakistan. Upon the completing of the degree, there is a grace period of up to six months before the loan must be repaid over a maximum of ten years. This program gives students access to inexpensive financing choices so they can pursue higher education and succeed in their careers. Critical analysis of the research item National Bank has a lot of resources so it can provide many students with funds since it is a bank, and many people deposit their saving in its accounts. The loan amount is PKR 2,000,000 which can cover the fees of most universities. Repayment of the loan can be done within 10 years which enables a student to work and repay the loan over time, students are not pressurized with short deadlines. The weaknesses are that there is a list of approved subjects which contains only the popular subjects and if the student is not studying any of them, he is not eligible for the loan, which is quite unfair as education in any subject is important. Moreover, Banks follow a business model return on Asset to ensure profitability, so their priority isn't helping students out their priority is ensuring profit for its lenders that is the reason why even in this loan scheme there is a heavy interest rate of 7% which is a great burden on the students who already have limited finances [4]. Has a strict eligibility criterion as it favors those candidates from whom it can expect repayment, resultantly many deserving candidates are left out, as one requirement says in the last public examination have must have secured 70% marks which could rule out a very deserving candidate if his last exam didn’t go well. Relationship to the proposed research work In terms of the scope of our website will offer things which are like this student loan scheme but will add upon them in a way which will make the process even better. The target demographic of our project and this scheme is quite similar as they both intend to facilitate students who are facing a difficulty in pursuing their education due to financial

circumstances. 2.2 Literature Review Summary Table Table 1: Comparison of Different Platforms The table shows comparison of the functionalities of different platforms. Features GoFundMe Akhuwat Foundation NBP Student Loan Scheme Scholar Support User Friendly Interface True False False False Login True False False True Chat True False False True Donation True True True Loan False False True True Share True False False True User Profile True False False True 2.3 Conclusion From the above review it is clear that a better solution to our problem is required critically and that would fill the gap in Pakistan's educational industry. Current organizations do not provide the power to choose who they want to provide money but with Scholar Support they can find a student whom they wish to help and in addition to that out integrated recommendation system would help them in that cause. Software Requirement Specifications The software requirements for Scholar Support are divided and elaborated in the below sections: 3.1 List of Features • Scholar Support will provide an interface where donors can provide donations for educational purposes. • Scholar Support will provide a platform where students can sign up and ask for donations. • Scholar Support will build a recommendation-based system where students will be recommended to donors on preference basis. • Scholar support will provide interface where students will be able to post on their timeline to ask for donations. • Scholar Support will maintain the record of the student loans or donations. 3.2 Functional Requirements The functional requirements of all the actors and system is thoroughly discussed under. 3.2.1 Functional Requirements of Admin • The system shall allow admin to accept or reject student's appeal for donation. • The system shall allow admin to remove a student from system. • The system shall allow admin to login using his email and password • The system shall allow admin to see student's documents attached with his/her request. 3.2.2 Functional Requirements for Students • The system shall allow student to create an account using his email and password. • The system shall allow student to login using his credentials. • The system shall allow student to edit his profile. • The system shall allow student to change his credentials. • The system shall allow student to post for donations and loans on his post. • The system shall allow student to upload his documents for verification. • The system shall allow student to upload his profile picture. • The system shall allow student to hide his identity by choosing 'stay anonymous' option. 3.2.3 Functional Requirements for Donor • The system shall allow donor to create an account using his email and password. • The system shall allow donor to login using his credentials. • The system shall allow donor to edit his profile. • The system shall allow donor to change his credentials. • The system shall allow donor to see donation requests by students. • The system shall allow donor to upload his profile picture. • The system shall allow donor to provide donation using his credit/debit card. • The system shall allow donor to see previous record of his donations or loans. • The system shall allow donor to receive his loans back from students. 3.2.4 Functional Requirements for System • The system shall use recommendation-based algorithm to automatically recommend the more needy students towards donors. • The system shall be able to keep record of all donations and loans provided to students. 3.3 Quality Attributes These are the following quality attributes of our system. 3.3.1 Security The system shall be able to protect itself from any kind of attack. 3.3.2 Compatibility The system shall be compatible with other systems and software. 3.3.3 Maintainability The system shall be able to repair and improve in future. 3.3.4 Performance The system shall be able to perform tasks in real time and respond to user queries. 3.4 Non-Functional Requirements The non-functional requirements of our system are as follows: 3.4.1 Security • The system shall provide access to the data that is protected by HTTP. • The system shall be able to protect itself from DOS attacks. • The system shall be able to encrypt all the passwords in database. 3.4.2 Compatibility • The system shall be compatible to any windows or iOS interfaces. • The system shall be able to cope with any environment. 3.4. 3 Performance • The system shall have the functioning time of less than 3 seconds. • The system shall start

functioning and show the interface in less than 4 seconds. • The system's Bounce Rate range should be between 30 to 40 percent. 3.5 Assumptions No special assumptions at this stage of project. 3.6 Hardware and Software Requirements Both hardware and software requirements are provided below: 3.6.1 Hardware Requirements A simple Computer System such as desktop or laptop where internet can be connected would be required. 3.6.2 Software Requirements • Visual Studio Code: IDE for coding • React JS: Front-End JavaScript Library • Node JS: Runtime environment • Python: General Purpose programming language • MongoDB: J like database • Express JS: Back-End web application framework 3.7 Use Cases All the use cases that would be essential for the proper functioning Scholar Support project are enlisted below: 3.7.1 Login Name Login Actors Student, Donor, Admin Summary The user shall provide their verified email address and 8-digit password on the login page and after successful verification, redirect the user to the home page. Pre-Conditions The user must be in the database records either added by any of the authorized users or added manually by a developer. The user must not already be logged in. Post-Conditions The user's session is successfully established and shall be redirected to the home page. Special Requirements None Basic Flow Actor Action System Response 1 The user opens the login page. 2 The login page is displayed asking for email and password. The system verifies the email and 3 The user enters valid email and password. 4 password, establishes a session for the user and redirects the user to the home page. Alternative Flow 3 The user enters invalid email or password. 4-A error message: Incorrect email or password The system responds with an entered. 3.7.2 Sign-up Name Signup Actors Student, Donor Summary Any person can sign up either as Donor or as Student. Pre-Conditions User should not be an existing member Post-Conditions User will become member of Student Support. Special Requirements None Basic Flow Actor Action System Response 1 User clicks on signup option on home page. 2 The system opens the signup page for user. 3 The user fills the signup form. 4 The system saves the users data in database. Alternative Flow 3 The enters non-verified email address. 4-A The system responds with a notification to user "please enter correct email". 3.7.3 Change Password Name Change Password Actors Student, Donor Summary Any user can change his password by visiting his profile. Pre-Conditions User should be logged in the system. Post-Conditions User's password will be updated in the database. Special Requirements None Basic Flow Actor Action System Response 1 The user clicks on his profile option. 2 The system opens the profile page for user. 3 The user clicks on the change password option. 4 The system displays a new password option. 5 The user enters a new password. 6 System saves the new password in database. No Alternative Flow 3.7.4 Forgot Password Name Forgot Password Actors Student, Donor Summary User can click on forgot password option while trying to log in. Pre-Conditions User should be an existing member. Post-Conditions User's password will be updated in the database. Special Requirements None Basic Flow Actor Action System Response 1 User clicks on forgot password option. 2 The system asks the user to enter valid email address. 3 The user enters the email address. 4 The system sends the link to change his password on the provided email address. Alternative Flow 3 The enters non-registered email 4-A The system responds with a notification to address. user "please enter registered email". 3.7.5 View Profile Name View Profile Actors Student, Donor Summary User can view his profile Pre-Conditions User should be logged in the system. Post-Conditions System will show user's profile. Special None Requirements Basic Flow Actor Action System Response 1 The user clicks on his profile button 2 Profile page for user where all his information is given. No Alternative Flow 3.7.6 Edit Profile Name Edit Profile Actors Student, Donor Summary User will be able to edit his profile. Pre-Conditions User be logged in the system. Post-Conditions Updated data will be saved in the database. Special Requirements None Basic Flow Actor Action System Response 1 The user clicks on his profile button. 2 The system opens the profile page for user where all his

information is given. 3 The user clicks on the edit profile option. 4 The system allows user to update his profile. No Alternative Flow 3.7.7 Student Post for Donation Name Student Post for Donation Actors Student Summary The student shall enter his requirement from the donors and press the post button on his timeline and after approval from the admin the post shall be displayed on donors' feed. Pre-Conditions The user must be in the database and must be logged in. Post-Conditions The donor will be able to see student's call for charity on his wall. Special None Requirements Basic Flow Actor Action System Response 1 The student writes his requirement on his wall and press on post. 2 The system saves the data and sends it to the admin. 3 The admin accepts the request. 4 Student's post is shown on donor's feed. Alternative Flow 3 The admin rejects the students appeal The system responds with a notification to for donation. 4-A student "Your Request was Denied" 3.7.8 Donor Accepts Student's Request Name Donor accepts student's donation request Actors Donor Summary The donor shall press the 'donate' or 'loan' button in response to student's post for donation. Pre-Conditions The student must have posted a request for loan or donation. Post-Conditions The student shall receive a notification of "loan approved by specific donor". Special Requirements None Basic Flow Actor Action System Response 1 The admin approves student's request. 2 The system saves the data and sends it to the donor's wall. 3 The donor clicks the loan or donate button. 4 Payment methods are shown on the screen No Alternative Flow 3.7.9 Donation Channel Integration Name Donation Channel Integration Actors Admin Summary The system should be integrated with payment channels for transactions. Pre-Conditions Donors should have a bank account. Post-Conditions None Special Requirements None Basic Flow Actor Action System Response 1 An API call is generated by the system to connect it to the bank. 2 A response from the bank is verified by system and a request is generated for the connection. 3 The donor enters the account details. 4 The system verifies the details and connects the account. Alternative Flow 3 The enters incorrect account details. 4-A The system responds with a notification to user "Incorrect details". 3.7.10 Email Option Integration Name Email Option Integration Actors Donor, Student Summary The donors should be able to send email to the student appearing on his wall. Pre-Conditions Student should have posted a loan or donation request. Post-Conditions Email sent. Special Requirements none Basic Flow Actor Action System Response 1 Donor selects the individual and clicks on the mail option. 2 A new tab opens to send an email. 3 The donor sends email. 4 The mail is sent to student and the tab closes. No Alternative Flow 3.7.11 Anonymous Option for Student Name Anonymous Option for Student. Actors Student Summary Students should be able to create an account without showing their original name. Pre-Conditions None Post-Conditions None Special Requirements None Basic Flow Actor Action System Response 1 The student clicks on his profile option. 2 Student's profile is shown by the system. 3 Student clicks on the anonymous option. 4 The system hides user's name from the donor's feed. No Alternative Flow 3.7.12 Remove Student Name Remove Student Actors Admin Summary The admin should be able to remove a student from Scholar Support. Pre-Conditions Student is registered in Scholar Support. Post-Conditions Student should no longer be registered in Scholar Support. Special Requirements None Basic Flow Actor Action System Response 1 Admin clicjs remove student option. 2 System shows a message "Are you Sure?". 3 Admin clicks on the "Yes" option. 4 The system deletes the student's record from the database. No Alternative Flow 3.7.13 Remove Donor Name Remove Donor Actors Admin Summary The admin should be able to remove a donor from Scholar Support. Pre-Conditions Donor should be registered in Scholar Support. Post-Conditions Donor should no longer be registered in Scholar Support. Special Requirements None Basic Flow Actor Action System Response 1 Admin clicks on the "remove" user option. 2 System shows a message "Are you Sure?". 3 Admin clicks on the "Yes" option. 4 The system deletes the donor's record from the database. No Alternative Flow 3.7.14 Loan Maintenance Name Loan Maintenance Actors

Donor, Student Summary The system should keep the record of all loans provided to students. Pre-Conditions Students must have some loan payments due. Post-Conditions System records the new transaction and updates the date. Special Requirements None Basic Flow Actor Action System Response 1 Donor chooses to provide loan to a student. 2 System triggers a timer for loan recovery payments. 3 4 The system sends an email to student to payback his loan ten days before the due date. 5 Student pays his loan installment. 6 The system updates the timer. No Alternative Flow 5 Student does not pay his installment. 6-A The system generates an email to admin for loan payment issue. 3.7.15 Documents Verification Name Documents Verification Actors Student, Admin Summary The attached documents are verified by admin. Pre-Conditions The student posts for donation request and attach documents. Post-Conditions The student's request is posted on donor's feed. Special Requirements None Basic Flow Actor Action System Response 1 Student posts for donation and attach documents with it. 2 The request is shown on pending page on admin's interface. 3 The admin verifies the attached documents and clicks on the accept option. 4 The post is posted on donor's feed. Alternative Flow 3 The admin clicks on the reject option. 4-A The request is rejected and a notification is sent to student. 3.7.16 Student's Post Acceptance Name Student's Post Acceptance Actors Student, Admin Summary The post of student is verified by admin. Pre-Conditions The student posts for donation request and admin after verification accepts that request to be shown on donor's feed. Post-Conditions The student's request is posted on donor's feed. Special Requirements None Basic Flow Actor Action System Response 1 Student posts for donation. 2 The request is shown on pending page on admin's interface. 3 The admin verifies the attached documents and clicks on the accept option. 4 The post is posted on donor's feed. Alternative Flow 3 The admin clicks on the reject option. 4-A The request is rejected and a notification is sent to student. 3.7.17 Donation History Name Donation History Actors Student Summary System keeps the record of donations takes by students. Pre-Conditions The student should have taken a lone at least once. Post-Conditions The amount of donation is added to the subtotal. Special Requirements None Basic Flow Actor Action System Response The system adds that amount to the 1 Student receive a donation from a donor. 2 previous donations taken by that particular student. No Alternative Flow 3.7.18 Loan Verification Name Loan Verification Actors Admin, Student Summary Student posts for loan request and admin verifies if the student has returned his previous loan. Pre-Conditions The student posts for loan request. Post-Conditions Loan is approved. Special Requirements None Basic Flow Actor Action System Response 1 Student posts for loan request. 2 System checks if the loan balance in student's history is null. 3 Admin approves the student for loan. 4 Loan amount is transferred to student. Alternative Flow 1 Student has some remaining loans on 2-A The request is rejected and a notification is him. sent to student to return previous loan first. 3.8 Graphical User Interface This section consists of the pictures of user interface of Scholar Support along with the functionality and use cases that they cover. Figure 1: Signup Page The new users will be able to sign-up as described in use case 4.7.2 Figure 2: Login Page This will be the first page that the general user would use to login to his account as described in use case 4.7.1 Figure 3: View Profile, Edit Profile This will be the user profile page where users can also edit his details as described in use cases 4.7.5 and 4.7.6 Figure 4: Landing Page This will be the landing page of Scholar Support which would be visible to both students and donors Figure 5: Student Feed This will be the student feed where student can post for donation request and attach documents as discussed in use case 4.7.7 Figure 6: Forgot Password User can change his password by putting his registered email address as discussed in use case 4.7.4 Figure 7: Donor's Feed This will be the donor's feed where all the posts from students would be visible Figure 8: Admin Dashboard This will allow admin to see all the students registered and their requests Figure 9: Donation Requests This will allow admin to accept or reject any donation

request posted by students 3.9 Database Design The database design consists of two parts. First being ER diagram that will consist of relations between the important entities in the project. Secondly data dictionary is also provided to let the user understand how the database will look like. 3.9.1 ER Diagram Figure 10: Entity Relationship Diagram for Scholar Support 3.9.2 Data Dictionary Table 2: Data Dictionary This table contains database's metadata Entity Attribute Data Type Nullable Field Length Constraint Description id String No 10 Primary Key Id of contribution Contributio n amount Decimal No 10 Not Null Amount donated Created At Time/Date No Not Null Date contributed Donor_id String No 10 Not Null Id of Donor Administrat or Admin_id String No 10 Not Null Id of Administrat or Donor_id String No 10 Primary Key Id of Donor Donor name String No 20 Not Null Name of Donor password String No 30 Not Null Password of donor account id String No 10 Primary Key Id of Payment amount Decimal No 10 Not Null Amount of Payment Payment status String No 10 Not Null Status of Payment Payment Payment method String No 10 Not Null Method of Payment Created at Time/Date No Not Null Date of Payment Contributio n String No 10 Not Null Id of Contributio n id String No 10 Primary Key Id of Profile title String No 100 Title in Profile Profile Ad Required amount Decimal No 10 Not Null Total amount required Is featured String No 10 Not Null Featured at top Current amount Decimal No 10 Not Null Total amount received Created at Time/Date No Not Null Date of creation of Profile End date Time/Date No Not Null Expiry date of profile id String No 10 Primary Key Id of comment Comments text String No 200 Text of comment emoji String No 10 Face drawn using keyboard id String No 10 Not Null Id of Application Student_Id String No 10 Not Null Id of Student status String No 10 Not Null Current status of the Application Application Submission date Time/Date No Not Null Date of Submission Review date Time/Date No Not Null Review Date Approval date Time/Date No Not Null Date Application was approved Created at Time/Date No Not Null Date it was created Student_id String No 10 Not Null Id of Student Student name String No 20 Not Null Name of Student password String No 30 Not Null Password of student 3.10 Risk Analysis Risk analysis of a web project involves finding, analyzing, and reducing risks that can effect the project negatively. Scholar Support might face both technical and business risks which are explained below. 3.10.1 Technical Risks Our project might face the risk of technological obsolescence, where the website's technology stack or infrastructure becomes outdated or unsupported, leading to compatibility issues or security vulnerabilities. Scholar Scope might also face the risk of emerging technologies or standards, such as new web browsers, devices, or programming languages, which may require updates or modifications to the website's code or infrastructure. 3.10.2 Business Risks Scholar Support might face the risk of changing consumer preferences and behavior, where the website may struggle to keep up with evolving user expectations, trends, or technologies, leading to a decline in user engagement or satisfaction. High-Level and Low-Level Design 4.1 System Overview The web platform will connect students with donors who are willing to pay for the student's tuition fee. Students, Donors and Administrators will login initially, after they have been registered. Students will post their detail and administrators would approve or disapprove it. The student and Donor will also have the option to communicate with each other using the chat feature. 4.2 Design Considerations We now put focus on design considerations of Scholar Support in this section. 4.2.1 Assumptions and Dependencies Following are the assumptions that we have taken while building this project and the dependencies present in it. Assumptions • Access to web browser and knowledge of how to use it. • Active Internet Connection. • Has an Operating System such as Windows, Linux, Mac OS installed. • Student is enrolled in a university. • Students on the website may be concerned of their privacy of their personal information. • Students and Donors from varied Demographics will access the webs platform. Dependencies Following are the dependencies present in the project: • Security • Payment Gateway • Legal Compliance • Domain Name • Web Hosting 4.2.2 General Constraints The constraints that

need to be followed for good design. Hardware and Software Environment Active internet connection is present. MERN stack must have browser compatibility with the browser. End User Environment Must be compatible with various screen sizes, internet connections and web browsers. Standards Compliance Standards must be complied with so we will comply with Total Validator Standard. Interoperability Requirements Interaction with other systems will be ensured so we will use APIs, conduct integration testing and writing proper documentation for interoperability requirements. Interface Requirements Those interfaces must be used which ensure compatibility between different systems. Data Formats must be followed, and industry standard protocols should be used such as HTTPs. Data Repository and Distribution Techniques MongoDB is used as a data repository in MERN stack projects. Efficient programming will be done which avoids memory leaks and introduces vulnerabilities in the software. Security testing will be conducted. Firewalls and access controls will also be used. Security techniques Sensitive data will be stored in encrypted form. Memory Limitations System should have a hard drive of at least 80 GB, memory of at least 4 GB and a processor not having less than 1.5 GHz clock speed. Performance Requirements Page load time and response time in around 10 seconds. Network Communications We will analyze the network traffic and monitor network performance. A wheel network will be used. Verification and Validation Testing criteria will be defined, and test cases will be used. The issues will be rectified. Selenium will be used for automated testing. 4.2.3 Goals and Guidelines • No student should leave their studies due to their financial condition. • To simplify the process of sending and receiving financial aid. • To use modern technologies to solve real world issues. • To deliver financial aid quickly. • Web Platform must be secure We will accomplish these goals by keeping our web platforms interface easy to use and reliable as applications built using React allow pages to load quickly. Code written will be well tested so there are no vulnerabilities and application are reliable. Consistency and transparency will be ensured in the application. Thus, providing a hassle-free experience to the users. 4.2.4 Development Methods The development method that will be used is Kanban as in such an application as great emphasis is given for flow of work. To optimize it would help teams to efficiently provide services. Kanban also emphasizes on continuous delivery meaning the release of incremental changes as in this project there is a need for frequent changes and improvements, so changing user needs and requirements are met properly. Kanban is suitable for a small team working on a project. Moreover, in this project close teamwork is required and Kanban has boards and cards which will help the teams to track the progress of the project. 4.3 System Architecture Three-tier Architecture is adopted. Interfaces are connected to the database by a layer which is referred to as Application Layer. Having separate layers enables us to reuse it across multiple modules in the application, furthermore it allows our application to be maintainable as changes can be easily made in one layer without affecting the other layers. Security is also increased as before adding to the database the input is validated by the application layer. Figure 11: High Level Architecture Diagram 4.3.1 Scholar Support System Module This module has numerous subsystems signup, login, verifying accounts and handling the money donated. Figure 12: Project's Component Diagram 4.3.2 Student Module The module has functionality for login which grants website access to the user who has been previously registered. Signup stores users in the database so that they can be given login access. A student can make his profile. He can post data to the application, furthermore he can edit and delete that data as well. Figure 13:Student Component Diagram 4.3.3 Donor Module The donor can also login and sign up after being verified and checked that the user exists in the database. He has been given the functionality to chat with the student as well where he can send messages and emojis. The donor can contribute funds as a donation or loan and they will be sent to the student after following a procedure. Figure 14:Donor Component Diagram 4.3.4 Administrator Module The administrator is responsible for the verification of

students and their accounts. He also manages the finances; he is also authorized to flag suspicious transactions. Figure 15: Admin Component Diagram 4.4 Architectural Strategies Our system architecture is designed using multiple strategies to achieve the desired functionality and efficiency. This section provides overview of the system's design, making sure that it fulfills the project requirements and can be implemented efficiently. 4.4.1 Programming Language As far as our designing is concerned, we will be using MERN stack framework. MERN stack follows a three-tier architecture, which consists of three layers. The presentation layer is concerned with user interface and will be developed using React JS which is a JavaScript library. Node JS is used at the backend. MongoDB which a document- oriented database will be used in data storage layer. 4.4.2 User Experience Design These are some of the points that we will follow to make user experience as good as possible. • The user should feel comfortable with the overall design of the system, including the color scheme, typography, and other visual elements. • Navigation should be intuitive and easy to understand, allowing the user to easily move through the system and find the information they need. • Interactive elements, such as buttons and forms, should be designed clearly and easy to use, allowing the user to accomplish their goals quickly and efficiently. • The system considers accessibility in mind, so people with disabilities can use the system effectively. • The system is expected to be responsive, adjusting itself to varying screen sizes and device types to provide a consistent user experience. • User testing should be conducted regularly in order to identify areas where the user experience can be improved. • Design decisions should be informed by user research, ensuring that the system is designed with the needs and preferences of the target user group in mind. 4.4.3 Future Plans for Extension We plan to make Scholar Support accessible to students across the globe. So, keeping this goal in mind, we'll be working on our project and its modules so that they can be extended whenever needed. The Software Design principles namely SOLID principles will be used for building this project. 4.5 Domain Model/Class Diagram The following is the class diagram of Scholar Support Project showing the interaction between different modules. Figure 16: Class Diagram 4.6 Sequence Diagrams The following are the sequence diagrams for Scholar Support application showing the exchange of information between different actors and modules. Figure 17: Login Sequence Diagram User wants to login to Scholar Support Figure 18: View Profile Sequence Diagram User wants to view his profile Figure 19: Sign-up Sequence Diagram A new user wants to create his account Figure 20: Academic Data Sequence Diagram Figure 21: Edit Profile Sequence Diagram User wants to edit his/her profile Figure 22: Scholarship Application Students wants to apply for scholarship in the institute Figure 23:Loan/Donation Request Student wants to apply for loan or donation from the donors Implementation and Test Cases Implementation and Test Cases 5.1 Implementation • Dashboard and Reporting: The prototype showcases a user-friendly dashboard for both students and donors. It provides summary information, such as the total funds raised, number of donations, and progress towards financial goals. It also generates reports to track donation history and impact. The prototype aims to provide stakeholders with a tangible demonstration of the Scholar Support Platform's functionalities and user experience. 5.1.1 Student Feed The given code is a React component that represents a feed page for a student. It utilizes pagination to fetch and display posts from a server. The component renders a container with a post creation form and a container that maps over the fetched posts to render individual feed items. Pagination functionality is implemented through state variables and functions to navigate between pages. Figure 24: Student feed Snapshot 5.1.2 Donors Feed The provided code is a React component called which represents a feed page for a donor. It imports various components and libraries. Finally, the Donor Feed component can be used elsewhere. Figure 25: Donors feed Snapshot 5.1.3 Mongoose Mongoose is a layer between the MongoDB and the NodeJS server that is used to communicate data and handle database

queries. Using mongoose, we will be defining the overall tentative schema of the project. This schema will have multiple tables that target multiple entities of the project. This code is a backend implementation using Express.js, Passport.js, and MongoDB. It sets up an Express server, establishes a database connection, and configures various authentication strategies (Google and Facebook). The server exposes several routes for different functionalities, such as handling news feeds, student information, donor information, and officer-related tasks. The code also includes a middleware for file uploads and defines a route for adding posts with an image. The server listens on port 5000. Figure 26: Mongoose Database Snapshot 5.1.4 Main Page This code is a React component that represents the main page of a web application. It imports necessary dependencies such as React and React Router. The component renders a header section with a background image and a title. It also includes sections for the student portal and the donor portal, each with a brief description and a button to login. The component is exported as the default export. Figure 27: Main feed Snapshot 5.1.5 User Management After Signup, they will be able to call the GET API which will in turn allow them to login and update information by calling the UPDATE API through the press of a button. 5.2 Test Case Design and Description Test cases will be employed to assess whether the system's use cases were all satisfied. These test cases will then be examined, and if there is a failure, the system's flaws or faults will be found and fixed. Test cases have two possible outcomes: passed or failed. If a test case is not finished within the allotted time frame for the software development life cycle, it will be marked as "Not Executed." 5.2.1 Anonymous Option for Student Test Case: Signup Page Test Case ID: 1 QA Test Engineer: Ibrahim Maaz Test case Version: 1.0 Reviewed By: Hadi Shehryar Test Date: 27/10/23 Anonymous Option Use Case Reference(s): for Student use case (4.7.11) Revision History: N/A Objective To check if the student can create account with anonymous identity. Product/Ver/Module: Student Module Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: N/A Step No. Execution description Procedure result Student writes anonymous in the username Students name appears as 1. section while creating an account anonymous in database and on donor’s wall. Comments: The test case is passed. System response was acceptable. Passed Failed Not Executed 5.2.2 View Profile Test Case: Profile Page Test Case ID: 2 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 28/10/23 Use Case Reference(s): View Profile use case (4.7.5) Revision History: N/A Objective To check if the student can view his profile. Product/Ver/Module: Student Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Student is logged into in his account. Step No. Execution description Procedure result 1. Student clicks on my account button Student’s account or profile is shown. Comments: The test case is passed. Passed Failed Not Executed 5.2.3 Edit Profile Test Case: Profile Page Test Case ID: 3 QA Test Engineer: Ibrahim Maaz Test case Version: 1.0 Reviewed By: Usman Elahi Test Date : 29/10/23 Use Case Edit Profile use case Reference(s): (4.7.6) Revision History: N/A Objective To check if the student/donor can edit their profile. Product/Ver/Module: Student and Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Student/Donor is logged into in his account. Step No. Execution description Procedure result 1. Student/donor clicks on my account button Student’s/Donor’s account or profile is shown. 2. User clicks on edit profile option. User’s data is shown 3. User enters new data and clicks on save Users updated data is saved in button. database. Comments: The test case is passed. Passed Failed Not Executed 5.2.4 Student Post for Donation Test Case: Profile Page Test Case ID: 4 QA Test Engineer: Ibrahim Maaz Test case Version: 1.0 Reviewed By: Hadi Shehryar Test Date: 05/11/23 Use Case Reference(s): Student Post for Donation use case (4.7.7) Revision History: N/A Objective To check if the student can request for donation on his profile. Product/Ver/Module: Student Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Student is

logged into in his account. Step No. Execution description Procedure result 1. Student clicks on request new donation button Request form is shown on student's profile. 2. Student fills the form and clicks the button. The request appears on admin profile and ultimately on donor's wall. Comments: The test case is passed. Passed Failed Not Executed 5.2.5 Donor Accepts Student's Request Test Case: Profile Page Test Case ID: 5 QA Test Engineer: Usman Elahi Test case Version: 1.0 Reviewed By: Hadi Shehryar Test Date: 05/11/23 Donor Accepts Use Case Reference(s): Student's Request use case (4.7.7) Revision History: N/A Objective To check if the donor can accept student's request on his profile. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Donor is logged into in his account. Step No. Execution description Procedure result 1. Donor clicks on donate button under Payment page is shown to the student's post. donor. Comments: The test case is passed. Passed Failed Not Executed 5.2.6 Student Login Test Case: Profile Page Test Case ID: 6 QA Test Engineer: Usman Elahi Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 07/11/23 Use Case Reference(s): Login use case (4.7.1) Revision History: N/A Objective To check if the student can log into his account. Product/Ver/Module: Student Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Student has already signed up for scholar support. Step No. Execution description Procedure result 1. Student enters valid email and password. Student's home page is shown to user. Comments: The test case is passed. Passed Failed Not Executed 5.2.7 Donor Login Test Case: Profile Page Test Case ID: 7 QA Test Engineer: Usman Elahi Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 09/11/23 Use Case Reference(s): Login use case (4.7.1) Revision History: N/A Objective To check if the donor can log into his account. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Donor has already signed up for scholar support. Step No. Execution description Procedure result 1. Donor enters valid email and password. Donor's home page is shown to user. Comments: The test case is passed. Passed Failed Not Executed 5.2.8 Admin Login Test Case: Profile Page Test Case ID: 8 QA Test Engineer: Usman Elahi Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date : 09/11/23 Use Case Reference(s): Login use case (4.7.1) Revision History: N/A Objective To check if admin can sign into his profile. Product/Ver/Module: Admin Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Admin was approved by another admin. Step No. Execution description Procedure result 1. Admin enters valid email and password. Admin's home page is shown to user. Comments: The test case is passed. Passed Failed Not Executed 5.2.9 Student Signup Test Case: Profile Page Test Case ID: 9 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 10/11/23 Use Case Reference(s): Signup use case (4.7.2) Revision History: N/A Objective To check if the user can create a new account as student. Product/Ver/Module: Student Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: N/A Step No. Execution description Procedure result 1. Student clicks on signup button. Signup page is shown. 2. Student enters the data in sign up form and clicks on submit button at the end of page. System saves data is database and shows login page. Comments: The test case is passed. Passed Failed Not Executed 5.2.10 Donor Signup Test Case: Profile Page Test Case ID: 10 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 10/11/23 Use Case Signup use case Reference(s): (4.7.2) Revision History: N/A Objective To check if the user can create a new account as Donor. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: N/A Step No. Execution description Procedure result 1. Donor clicks on signup button. Signup page is shown. 2. Donor enters the data in sign up form and clicks on submit button at the end of page. System saves data is database and shows login page. Comments: The test case is passed. Passed Failed Not Executed 5.2.11 Student Change Password Test Case:

Profile Page Test Case ID: 11 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 12/11/23 Use Case Change Password use Reference(s): case (4.7.3) Revision History: N/A Objective To check if the student can change his password. Product/Ver/Module: Student Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Student is logged into his account. Step No. Execution description Procedure result 1. Student clicks on the change password button on his edit profile page. System shows new page to change password. 2. Student enters the new password. New password is saved in database. Comments: The test case is passed. Passed Failed Not Executed 5.2.12 Donor Change Password Test Case: Profile Page Test Case ID: 12 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Usman Elahi Test Date: 17/11/23 Use Case Reference(s): Change Password use case (4.7.3) Revision History: N/A Objective To check if the donor can change his password. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Donor is logged into in his account. Step No. Execution description Procedure result 1. Donor clicks button to change password on his edit profile page. System asks for old password and to enter new password if the old password matches, in a new page. 2. Donor enters the new password. New password is saved in database. Comments: The test case is passed. Passed Failed Not Executed 5.2.13 Student Forgot Password Test Case: Profile Page Test Case ID: 13 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Usman Elahi Test Date: 20/11/23 Use Case Forgot Password use Reference(s): case (4.7.4) Revision History: N/A Objective To check if the student can change his password by using forgot password option. Product/Ver/Module: Student Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Student is already a member od scholar support. Step No. Execution description Procedure result 1. Student clicks on the forgot password A new page is shown that requests button on sign in page. an email and sends the rest link. Comments: The test case is passed. Passed Failed Not Executed 5.2.14 Donor Forgot Password Test Case: Profile Page Test Case ID: 14 QA Test Engineer: Ibrahim Maaz Test case Version: 1.0 Reviewed By: Usman Elahi Test Date: 21/11/23 Use Case Forgot Password use Reference(s): case (4.7.4) Revision History: N/A Objective To check if the donor can change his password by using forgot password option. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Donor is already a member of scholar support. Step No. Execution description Procedure result 1. Donor clicks on the forgot password button on sign in page. A new page is shown that requests an email and sends the rest link. Comments: The test case is passed. Passed Failed Not Executed 5.2.15 Donation Channel Integration Test Case: Profile Page Test Case ID: 15 QA Test Engineer: Usman Elahi Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 12/11/23 Donation Channel Use Case Reference(s): Integration use case (4.7.9) Revision History: N/A Objective To check if the donor can donate money to student. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Donor is logged into in his account. Step No. Execution description Procedure result 1. Donor clicks on donate button under student's post. Payment page is shown to the donor. 2. Donor enters his card details to send money. Money is transferred to student. Comments: N/A. Passed Failed Not Executed 5.2.16 Email Option Integration Test Case: Profile Page Test Case ID: 16 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 15/11/23 Use Case Reference(s): Email Option Integration use case (4.7.10) Revision History: N/A Objective To check if the donor can send email to student. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Donor is logged into in his account. Step No. Execution description Procedure result 1. Donor clicks on email button under student's post. Email page is shown to donor. Comments: The test case is

passed. Passed Failed Not Executed 5.2.17 Remove Student Test Case: Profile Page Test Case ID: 17 QA Test Engineer: Usman Elahi Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 05/11/23 Use Case Remove Student use Reference(s): case (4.7.12) Revision History: N/A Objective To check if the admin can remove a student from database. Product/Ver/Module: Admin Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Admin is logged into in his account. Step No. Execution description Procedure result 1. Admin clicks on Students page on his dashboard. System shows him all the registered students. 2. Admin clicks on the remove student button under the student he wants to remove. Student is removed from the web portal's database. Comments: The test case is passed. Passed Failed Not Executed 5.2.18 Remove Donor Test Case: Profile Page Test Case ID: 18 QA Test Engineer: Usman Elahi Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 05/11/23 Use Case Remove Donor use Reference(s): case (4.7.13) Revision History: N/A Objective To check if the admin can remove a donor from database. Product/Ver/Module: Admin Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Admin is logged into in his account. Step No. Execution description Procedure result 1. Admin clicks on Donors page on his dashboard. System shows him all the registered donors. 2. Admin clicks on the remove donor button under the donor he wants to remove. Donor is removed from the web portal's database. Comments: The test case is passed. Passed Failed Not Executed 5.2.19 Document Verification Test Case: Profile Page Test Case ID: 19 QA Test Engineer: Ibrahim Maaz Test case Version: 1.0 Reviewed By: Hadi Shehryar Test Date: 05/11/23 Use Case Reference(s): Document Verification use case (4.7.15) Revision History: N/A Objective To check if the admin can verify documents submitted by students. Product/Ver/Module: Admin Module. Environment: Internet working and browser working. Assumptions: Admin is logged into in his account. Pre-Requisite: Student has applied for a new donation. Step No. Execution description Procedure result 1. Admin selects the view option under the students request. Supporting documents are shown in a separate page. 2. Admin clicks the accept option on the student's request Post is accepted and status is changed in database. Comments: The test case is passed. Passed Failed Not Executed 5.2.20 Donation History Test Case: Profile Page Test Case ID: 20 QA Test Engineer: Hadi Shehryar Test case Version: 1.0 Reviewed By: Ibrahim Maaz Test Date: 05/11/23 Use Case Donation History use Reference(s): case (4.7.17) Revision History: N/A Objective To check if the donor can view his donation history. Product/Ver/Module: Donor Module. Environment: Internet working and browser working. Assumptions: N/A Pre-Requisite: Donor is logged into in his account. Step No. Execution description Procedure result 1. Donor clicks on the donation history page. His donation his shown in a page. Comments: The test case is passed. Passed Failed Not Executed 5.3 Test Matrices 5.3.1 Test Case Matrix Metrix Purpose No of Test Cases 20 No of Test Cases Passed 13 No of Test Cases Failed 7 Test Case Defect Density 0 Test Case Effectiveness 0 Traceability Matrix Submitted in a separate file User Manual In this segment, we present the user manual for Scholar Support. This manual is intended to successfully direct clients through the functionalities and highlights of our undertaking, offering definite guidelines on the best way to utilize it. We plan to give clear and brief data to help you explore and saddle the maximum capacity of our venture. Along these lines, how about we make a plunge and investigate how Scholar Support can work on your errands and improve your experience. 6.1 Sign-up • Go to the homepage of the scholar support website. • Look for "Create Account" or "Sign Up" button. • Enter all required data accurately, including your email address, password, and username. • Observe any instructions or prerequisites listed on the sign-up page. • Verify the password and supply any more details that may be required. 6.2 Login • Go to the scholar support website. • On the homepage, look for the "Login" button. • Press the corresponding button to be taken to the login screen. • Enter your username or email address that you have on file in the

relevant field. • In the password field, type your password. • Make sure you type accurately to prevent login problems. 6.3 Edit Profile • Go to the settings of your profile. • Search for a "Edit Profile" option. • Examine and amend personal information, including name, contact details, and any other pertinent fields. • After making changes to your profile, save the changes. 6.4 Forgot Password • Navigate to the scholar support login page. • Search for the link labeled "Forgot Password." • Enter the email address linked to your account on the password recovery page. • To receive instructions on how to reset your password, make sure the email address is correct. 6.5 Student Request for Donation • Sign in to your account. • Click on the "Request a New Donation" option. • Fill the required fields. In this segment, we present the user manual for Scholar Support. This manual is intended to successfully direct clients through the functionalities and highlights of our undertaking, offering definite guidelines on the best way to utilize it. We plan to gi • Click on the "Submit" option. 6.6 Anonymous Option for Student and Donors • Go to the homepage of the scholar support website. • Look for "Create Account" or "Sign Up" button. • Enter "Anonymous" in the username field. 6.7 Remove Student • Admin sign in on his account. • Go to the Students page. • Select the "Remove" option under the student you want to remove from scholar support web portal. 6.8 Remove Donor • Admin sign in on his account. • Go to the Donors page. • Select the "Remove" option under the donor you want to remove from scholar support web portal. 6.9 Documents Verification • Go to the admin dashboard page. • Select the admin actions page. • Under student's requests click on the "view" option of supporting documents. • If verified click on the "Accept" option. 6.10 Donor's Donation History • Donor signs in on his account. • Click on the "Donation History" option on his profile. 6.11 Donation Acceptance by Donor • Donor signs in on his account. • Requests of students appear on his wall. • Donor clicks on the "Accept" option under the student he wants to donate. • Payment page will open. • Donor puts his bank details to donate. Conclusion and Future Work The scope of the project is explained in this deliverable. Our goal was to make an AI integrated website that will help students get donations from donors which we have accomplished. By literature review we have identified that there are already some organizations that are working for that cause but there is a gap in the market that will be fulfilled by Scholar Support. The high-level and low-level design of the project is also mentioned along with class diagram and SRS diagram is also shown that will help the reader understand the relations of modules along with their working. The sequence diagrams have also been drawn to further provide ease to the reader to understand our project. The functionalities such as AI integration which will rank the student profiles based on their relevance to the donors and the payment processing feature have been integrated. Backend was developed using Node.js. Some other minor functionalities such as ability to exchange mails was added. Unit testing of crucial functions was performed along with Performance testing of the entire system using Puppeteer and Mocha. Test cases were created, executed, analyzed and maintained to make the application robust and reliable. References References [1]"GoFundMe's 2021 Giving Report," GoFundMe. https://www.gofundme.com/c/gofundme- giving-report-2021. [Accessed 01 March 2023]. [2] S. Ovide, "The Inequality of the GoFundMe Economy," The New York Times, Jun. 21, 2021.https://www.nytimes.com/2021/06/21/technology/gofundme-online-giving.html. [Accessed 03 March 2023]. [3] "Interest Free Loans | Akhuwat USA | Interest Free Microfinance," www.akhuwat.org. https://www.akhuwat.org/interest-free-loans [Accessed May 25, 2023]. [4] D.Puri, "Student loans used for exploitation," The Democracy Labs. https://thedemlabs.org/2021/12/31/student-loans-used-by-banks-to-exploit-students/ [Accessed 10 March 2023]. Scholar Support 3 Scholar Support 5 Scholar Support 7 Scholar Support 9 Scholar Support 11 Scholar Support 13 Scholar Support 15 Scholar Support 17 Scholar Support 19 Scholar Support 21 Scholar Support 23 Scholar Support 25 Scholar Support 27 Scholar Support 29 Scholar Support 31 Scholar Support 33 Scholar Support 35 Scholar Support 37

Scholar Support 39 Scholar Support 41 Scholar Support 43 Scholar Support 45 Implementation and Test Cases Scholar Support 47 Implementation and Test Cases Scholar Support 49 Implementation and Test Cases Scholar Support 51 Implementation and Test Cases Scholar Support 53 Implementation and Test Cases Scholar Support 55 Implementation and Test Cases Scholar Support 57 Implementation and Test Cases Scholar Support 59 Implementation and Test Cases Scholar Support 61 Implementation and Test Cases Scholar Support 63 Scholar Support 65