

WEB-BASED SCHOLARSHIP GRANT MANAGEMENT SYSTEM FOR THE
SANGGUNIANG KABATAAN OF GATID STA. CRUZ, LAGUNA

A Final Project
Presented to the
Faculty of College of Computer Studies
Laguna State Polytechnic University
Santa Cruz Campus

In partial Fulfilment of the Requirements for the course

CLIENT SERVER TECHNOLOGIES

Submitted by:

Lance Lordvin Concio

Mar Aldrei Y. Avila

Jake Paul J. Valenzuela

Jami Hyari Andaya

December 2025

CHAPTER I

INTRODUCTION

The Sangguniang Kabataan (SK) of Barangay Gatid, Santa Cruz, Laguna has a scholarship and grant program for students. Currently, the application and monitoring of scholarships is still done manually. Students still have to go to the SK Gatid office just to submit the documents. SK officials also review each application one by one, which makes the process time-consuming. Therefore, this study wants to create a web-based system for SK Gatid scholarships and grants. With this system, it is expected that the application process will be easier and faster, and more students will benefit.

The current scholarship and grant system at SK Gatid has problems. One of them is the lengthy process, which is a hindrance to students. In addition, it is possible for papers submitted by Gatid scholars to be lost and for errors to be made in recording information. Because of these problems, we researchers thought of creating a better system to improve SK Gatid's services.

Today, many organizations use web-based systems for their scholarships and grants. In various places, this technology has been proven to help speed up the process, reduce errors, and make management more transparent. The use of client-server technology allows for faster access and better data management.

Background of the Study

The use of client-server technology greatly contributes to the speed and efficiency of the systems. In the client-server model, clients (such as students and SK officials) can access information and services from a central server. This is faster and more reliable than the manual process, which requires going to the office and submitting documents.

Client-server technology is important in this project because it provides a central database where all information about scholarships and grants can be stored and managed. This allows students to apply online, and SK officials to track applications and generate reports. In addition, client-server technology allows for a higher level of security and data protection.

Research Problem

The current scholarship and grant management process of SK Gatid, Santa Cruz, Laguna is still done manually. Students still have to go to the SK office to submit their requirements and forms. SK officials check and process each application one by one, which slows down the process. This leads to delays, recording errors, and sometimes loss of documents. Due to these problems, students are struggling and sometimes they do not receive the financial assistance they are entitled to.

The main reasons for this problem are the lack of an online or centralized system for applying and monitoring scholars. Since the process is still paperbased, the processing of applications is slow, it is difficult to check

beneficiaries, and the evaluation is complicated. If this is not resolved immediately, the smooth flow and integrity of the SK scholarship program will be affected.

Based on observations and discussions with several SK officials and students, there are instances where documents are lost and applications take a long time to be approved. Students also have difficulty knowing the status of their applications. This shows that there is a dire need for a system that can help make the scholarship management process faster, more efficient, and easier.

To address these problems, this study proposes the creation of a webbased management system for SK Gatid scholarships and grants. Through this system, the process of applying, selecting, and monitoring scholars will be easier and more efficient.

Specific Problems:

1. How can this system help speed up the scholarship application process?
2. How can the system reduce errors and loss of student documents?
3. How can the system help SK officials in monitoring and managing scholars?
4. How can the system increase transparency and accessibility for students who want to apply for scholarships?

Project Objective

The purpose of this research is to develop a system for the Management of SK Gatid Scholarships and Grants to solve the problems in the current manual process and to expedite the application, selection, and monitoring of scholars.

specifically it aims to:

1. Create a user-friendly website for SK Gatid students and officials.
2. Create a secure database to store information about Gatid applicants and scholars.
3. Automate the process of evaluating applications and selecting beneficiaries.
4. Create reports to see if the program is performing well and to help in decision-making.
5. Test the system to ensure it is working properly and is helpful to users.

Scope and Limitation

This study is about the development of a web-based system for scholarships and grants of SK Gatid, Santa Cruz, Laguna. It aims to facilitate and speed up the application process of students and the processing of SK officials. The system includes online application, checking requirements, monitoring scholars, and creating reports to better run the program. This system will be used by SK officials and students from Barangay Gatid who want to apply for scholarships.

This study is limited to residents of Barangay Gatid and cannot be used by students from other barangays. An internet connection is also required to

use the system because it is web-based. The system also depends on the correct entry of information by users, so if they enter something incorrectly, that is not the responsibility of the system developers. Maintenance and future updates of the system after its creation are not included in the study.

Significance of the Study

The system will help the SK officials of Barangay Gatid because they will be able to manage scholarships and grants faster and more accurately. It will also help the students of Barangay Gatid because it will be easier for them to apply and track the status of their applications.

Potential Users:

SK Officials of Barangay Gatid: They will directly manage and use the system for processing applications and tracking scholars.

Student residents of Barangay Gatid: They are the applicants and will benefit from the easier process of applying for scholarships.

CHAPTER II

SYSTEM ANALYSIS AND DESIGN

Overview of Client-Server Model

The client-server model is a system design in which the functions of the software are divided into two parts: client and server. In this project, which is the web-based scholarship and grant management system of SK Gatid, this model is used to better and faster manage the information of students and SK staff.

Client

This is the part of the system that is directly used by students or SK staff. For example, this is where students apply for scholarships, and this is where they can also see the status of their application. The client sends a request to the server and displays the result to the user.

Server

This is the part that processes all requests from the client. This is where all the data is stored, such as applicant information and application status. The server also ensures that all the information is correct and secure before returning it to the client.

The two interact over the network. When the client sends a request, the server processes it and sends the result back to the client.

Benefits of the client-server model in system design:

Centralized management: All data is on the server, making maintenance and updates easier.

Scalability: Multiple users can access it simultaneously.

Security: Information is more secure because it is stored on the server.

Efficiency: Roles are organized, client for interface and display, server for processing and storage.

System Requirements

This section specifies the hardware, software, and network requirements for the SK Gatid Scholarship & Grant Web-Based Management System. These requirements ensure that the client side (students and SK officials) and the server side can operate the system properly.

Hardware Requirements

Client-Side

Device Type: Desktop, laptop, tablet, or smartphone

Processor: Dual-Core 1.8 GHz or equivalent

RAM: 8GB minimum

Storage: 50GB available space

Peripheral: Camera or scanner (for uploading documents)

Server-Side

Device Type: Desktop computer or local server

Processor: Quad-Core 2.0 GHz or higher

RAM: 8GB or higher

Storage: 250GB minimum (for system files, database, and uploads)

Server Hosting: If online, server/hosting must support PHP & MySQL

Software Requirements

Client-Side

Web Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari

Server-Side

PHP Version: PHP 7.4 or higher

Database System: MySQL 5.7+

Web Server: Apache (included in XAMPP)

Local Development Tool: XAMPP

Database Connection: PDO (PHP Data Objects)

Email Sending: PHPMailer 6.12.0 (for Gmail SMTP)

Front-End Tools

HTML Version: HTML5

CSS Version: CSS3

JavaScript Version: ES6+

Frameworks & Libraries: Bootstrap 5.3.3, Font Awesome, Chart.js

API Dependencies

Google Gemini API: AI assistance for form filling and suggestions

IP Geolocation API: Checks applicant location

Public IP API: Gets user's IP address

Google Maps JavaScript API: Displays map and SK office location

Gmail SMTP: Sends automated emails

1. Network Requirements

Internet Requirements

Stable Internet for: accessing the system, API calls, submitting forms, uploading documents, sending emails

Minimum Speed: 5 Mbps

Local Network (if used LAN-based)

Router: With DHCP enabled

Connection: LAN cable or Wi-Fi

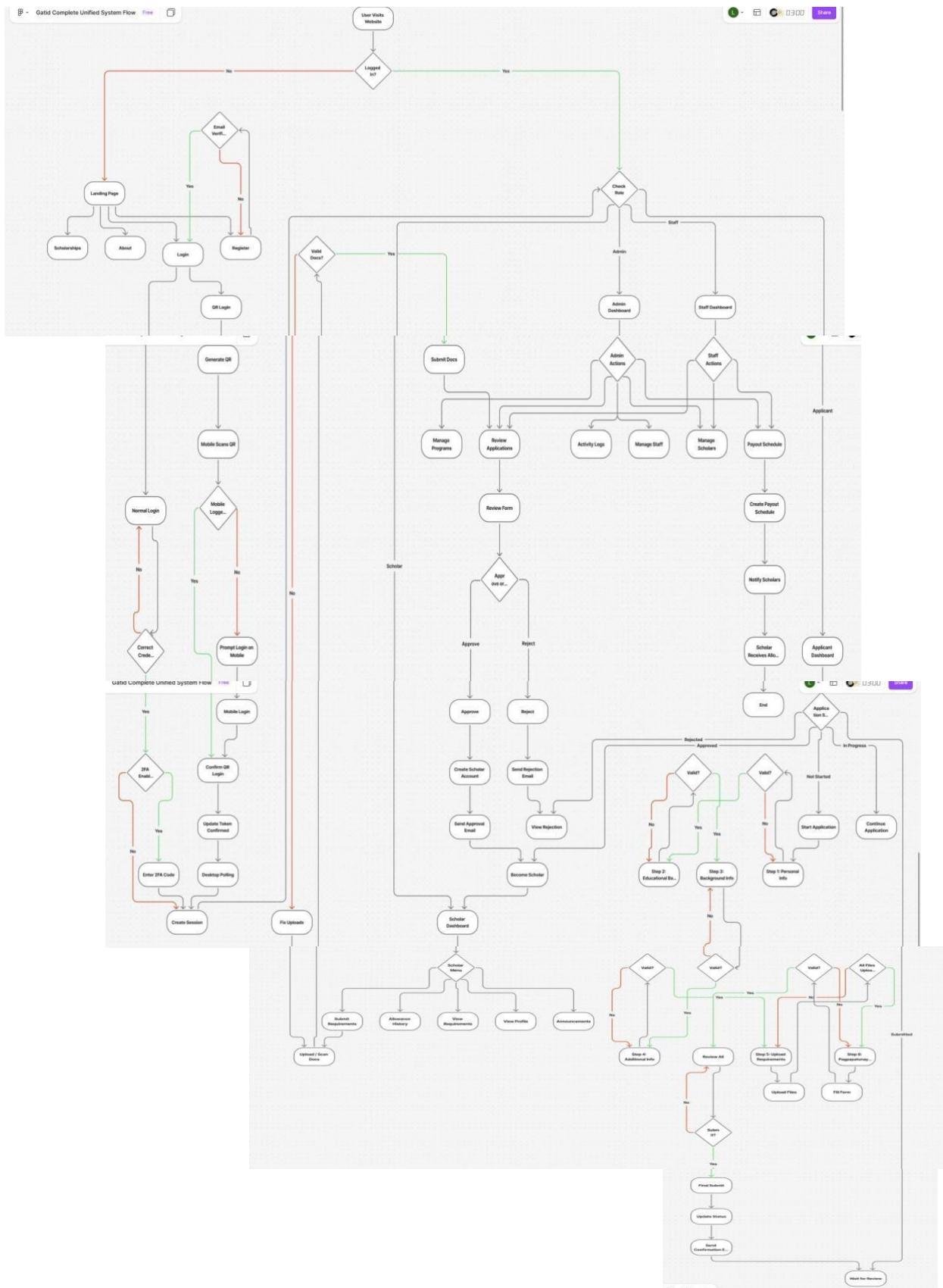
Network Protocols

HTTP/HTTPS: For secure client-server communication

TCP/IP: Standard data transmission protocol

SMTP: Sending emails through Gmail

System Design Flowchart



API Integration

The SK Gatid Scholarship & Grant Web-Based Management System uses several external APIs to enhance system features, automate certain processes, and assist users during the application process. These APIs provide AI assistance, accurate location analysis, automatic email sending, and map display. The main APIs used are:

1. Google Gemini API

Purpose: Provides AI support through a chatbox or virtual assistant for students applying. It helps the user fill out the form, answer questions, and provide suggestions to avoid mistakes. This makes the application process easier and faster.

Functionality: There are endpoints for analyzing text and creating content or suggestions.

Data Exchange: Text request to JSON AI response.

Integration Method: PHP cURL using REST API endpoint.

Request Type: POST

Endpoint: /v1beta/models/gemini-pro:generateContent

Authentication: API key

2. IP Geolocation API

Purpose: Find out where the user is applying to ensure that the applicants are only from Barangay Gatid.

Functionality: Provides information such as city, region, and country based on IP address.

Data Exchange: IP to JSON location data.

Integration Method: PHP GET request via `file_get_contents()` or cURL.

Request Type: GET

Endpoint: /json

3. Public IP Detection API

Purpose: Get the user's public IP, which the Geolocation API will use to determine location.

Functionality: Provides the client's IP address.

Data Exchange: GET to JSON (IP).

Integration Method: PHP GET request.

Request Type: GET

Endpoint: /?format=json

4. Gmail (SMTP) Simple Mail Transfer Protocol API

Purpose: Send automated emails to students and SK officials regarding application status and other announcements.

Functionality: Uses Gmail SMTP server to send email.

Data Exchange: Email fields to Gmail SMTP to user's inbox.

Integration Method: PHPMailer + SMTP settings.

Request type: SMTP Commands (HELO/EHLO, MAIL FROM, RCPT TO, DATA)

Endpoint: smtp.gmail.com

Authentication: App password

5. Google Maps API

Purpose: Show the map and directions, such as the location of Barangay Gatid and SK office.

Functionality: Provides maps, directions, geocoding, and place services.

Data Exchange: Map request to map display.

Integration Method: JavaScript on the front-end.

Endpoint: Google Maps JavaScript API

Authentication: API key

Data Flow and User Interface

This section illustrates how data flows through the SK Gatid Scholarship & Grant Web-Based Management System and presents the basic user interface design that users interact with. It explains how data moves from user input to server processing and back, as well as the typical user journey through the system.

Data Flow

The movement of data in the system can be summarized in the following steps:

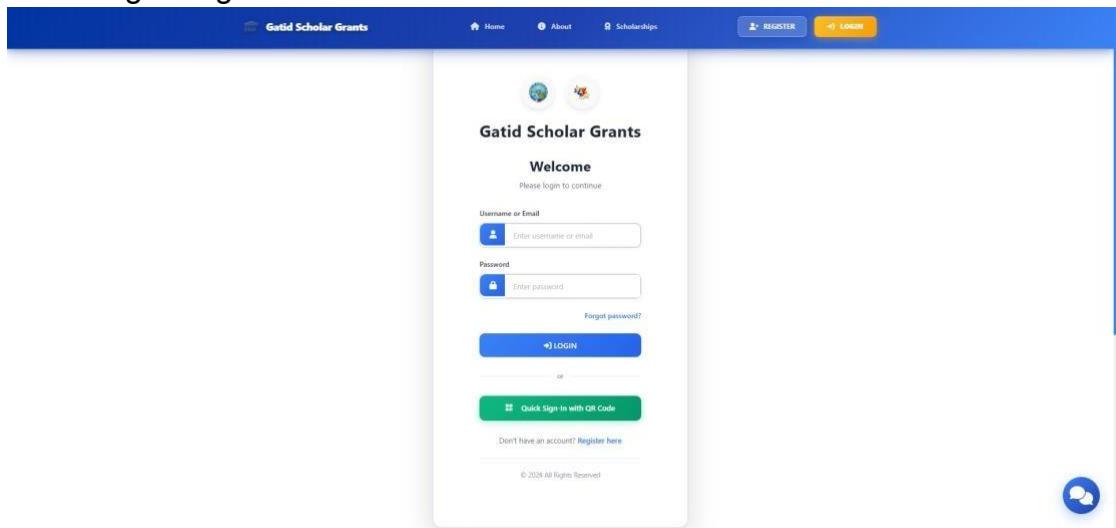
1. Login – Users enter their username and password.
2. Dashboard Access – Users access their respective dashboard (Student or Official).
3. Form Submission – Students fill out and submit the scholarship application form.
4. Server Processing – The server validates, stores, and retrieves information from the database.
5. API Interaction – External APIs are called if needed (e.g., Google Gemini, Geolocation API, Gmail SMTP, Google Maps).
6. Result Display – The processed data is returned to the client and displayed to the user.

Typical User Journey

1. Login Page – Users securely log in to access the system.
2. Dashboard – Students view application status; SK officials manage applications.
3. Application Form – Students fill in required details and upload supporting documents.
4. Evaluation & Monitoring – SK officials review, approve, or reject applications and track scholars.
5. Notifications & Maps – System provides email notifications and displays SK office location.
6. Logout – Users securely log out of the system after completing their tasks.

Key Interface Pages

1. Login Page

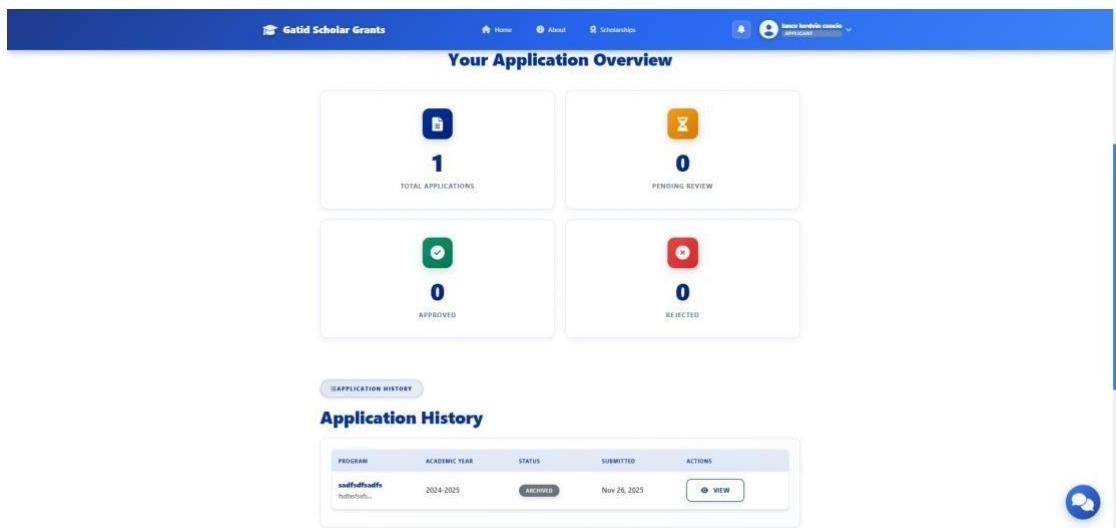


The login page for Gatid Scholar Grants features a blue header with the website name. Below the header is a central form area with a "Welcome" message and a "Please login to continue" note. It includes fields for "Username or Email" and "Password", both with "Enter" buttons. There are links for "Forgot password?", "LOGIN", and "Quick Sign-in with QR Code". At the bottom, there's a "Don't have an account? Register here" link and a copyright notice. A blue circular icon with a white speech bubble is in the bottom right corner.

Figure 1:

Description: Users log in here to access their dashboard.

2. Student Dashboard



The student dashboard is titled "Your Application Overview". It displays four summary boxes: "TOTAL APPLICATIONS" (1), "PENDING REVIEW" (0), "APPROVED" (0), and "REJECTED" (0). Below this is a section titled "Application History" with a table showing one application entry. The table columns are PROGRAM, ACADEMIC YEAR, STATUS, SUBMITTED, and ACTIONS. The entry shows "softsoftsofts", "2024-2025", "ARCHIVED", "Nov 26, 2025", and a "VIEW" button. A blue circular icon with a white speech bubble is in the bottom right corner.

Figure 2:

Description: Students can see their application status and submit forms.

3. Official Dashboard

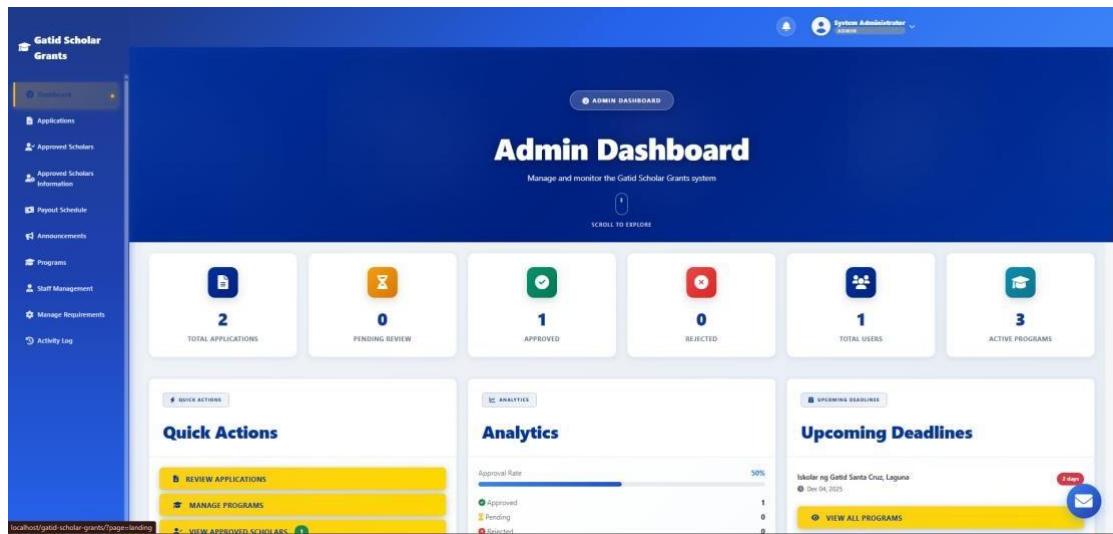


Figure 3:

Description: Officials manage and monitor all student applications.

4. Application Form Page

Figure 4:

Description: Students fill out this form and submit required documents.

5. Reports Page

The screenshot shows the 'Approved Scholars Information' page. At the top, there's a warning message: 'Warning: This page allows you to reset ALL information for approved scholars (including personal info, educational background, background information, additional information, requirements, and documents). This action cannot be undone!' Below this, there are three summary boxes: '1 SCHOLARS', '1 COLLEGES', and '1 PROGRAMS'. A search bar for 'Search scholars...' is present. The main section is titled 'List of Approved Scholars' and displays a single entry for 'Iskolar ng Gatid Santa Cruz, Laguna'. The table columns include SLOT #, NAME, CONTACT, SCHOOL, COURSE, YEAR LEVEL, ACADEMIC YEAR, APPROVED DATE, and ACTIONS. The ACTIONS column contains a blue envelope icon.

Figure 5:

Description: Officials can view reports of student applications and monitor program performance.

6. Map Page

The screenshot shows the 'Our Location' page. It features a heading 'Gatid Barangay Hall' with the address '7994+999, Santa Cruz, Laguna'. Below the heading is a map of the area around Gatid Barangay Hall, showing nearby landmarks like 'Gatid House', 'Gatid House Inspired Lodge and Motel', and 'Argent Condominium'. A red marker indicates the location of Gatid Barangay Hall. A 'GET DIRECTIONS' button is at the bottom of the map. The page also includes a 'VISIT US' button and a 'Find us at the Gatid Barangay Hall' link. A blue speech bubble icon is in the bottom right corner.

Figure 6:

Description: Shows the location of SK office for student reference.

CHAPTER III

IMPLEMENTATION

Technology Stack

The system was built using various programming languages, tools, and software. These helped to build the web-based management system of the SK Gatid Scholarship and Grant Program. The technologies used and what they are for are listed below.

FRONT-END

1. HTML5 (Hypertext Markup Language)

This is used to build the basic structure of our website pages, like the login page, application forms, and dashboards.

2. CSS3 (Cascading Style Sheets)

This helps us style the website, so we can change colors, fonts, and the layout to make it look good.

3. JAVASCRIPT (ES6+)

This makes the website interactive. It's used for things like making buttons work and checking if the forms are filled out correctly.

4. BOOTSTRAP 5.3.3

This is a tool that helps us design the website quickly and make sure it looks good on both phones and computers.

5. FONT AWESOME 6.0

Used for icons to make the website more visually appealing.

6. CHART.JS 3.9.1

Used to display data in charts, making it easier to understand.

BACK-END

1. PHP 7.4+ (Hypertext Preprocessor)

This is the main language we used to process data and connect to the database.

2. MySQL 5.7+

This is the database where all the information about students, applications, and scholars is stored.

3. PDO(PHP Data Objects)

Used to help PHP connect to the database.

4. PHPMailer 6.12.0

Used to send emails from the system, like application updates.

SERVER

1. XAMPP

This is a local server we used while building and testing the system. It includes PHP and MySQL, so we can run the website on our computers. 2. Apache

Included in XAMPP, Apache is the web server that handles requests and serves the website.

System Modules

To make our system easy to manage and update, we've divided it into several functional modules. Each module is responsible for specific tasks within the system. Here's a breakdown:

Module	Description of Role and Key Functions
Scholarship Application Module	Handles the scholarship application process.
Applicant Evaluation Module	Manages the evaluation of scholarship applicants.
Scholar Management Module	Manages information about current scholars.
Reporting and Analytics Module	Generates reports and provides data insights.

		and the overall implementation of the scholarship program.
User Authentication and Authorization Module	Ensures secure access to the system.	Manages user accounts, login credentials, and access permissions for different user roles (e.g., students, SK officials).
Content Management Module	Manages the content displayed on the website.	Allows administrators to update announcements, scholarship guidelines, and other important information.
Communication/Notification Module	Facilitates communication with applicants and scholars.	Sends automated email and SMS updates regarding application status, upcoming deadlines, and important announcements.

Document Management Module	Manages the storage and retrieval of documents.	Securely stores uploaded application documents, scholar records, and other relevant files.
Feedback and Support Module	Provides a platform for feedback and support.	Allows students to submit inquiries, report issues, and receive assistance from SK officials.

Testing and Debugging

This section explains how the system was tested and how errors were fixed during development. Different tests were done to make sure that every feature worked properly, and debugging tools were used to find and solve problems in the system.

Testing Approaches

Unit Testing

Unit testing was done to check small parts of the system separately.

Examples include:

Validating login inputs

Testing form submission functions

Checking email sending

Checking API responses.

Integration Testing

Integration testing checked if different modules work correctly when combined.

This includes:

Saving submitted forms to the database

Displaying student records on the admin dashboard

Sending email notifications

Receiving IP and location data from APIs.

Manuall Testing

Manual testing was done by using the system as a normal user.

This helped identify errors such as:

Incorrect outputs

UI issues

Missing alerts

Broken links or buttons.

User Acceptance Testing (UAT)

Selected users such as SK officials and students tested the system to check if:

The interface is easy to use

The instructions are clear

The features match their needs

Debugging Tools

PHP Error Reporting: used to display warnings and errors in PHP code.

Xdebug: used to trace the code step-by-step and locate the exact line of error.

Chrome Developer Tools: used for checking JavaScript errors, CSS layout issues, and failed API calls.

phpMyAdmin Debugging: used for testing SQL queries and checking database problems.

API Testing (Manual): manual checking of network logs to verify if API calls return correct data.

Sample Test Cases

Test Cases	Expected Output	Actual Result	Status
Login with correct username and password	Redirects to Dashboard	Works as expected	Passed
Submit scholarship form	Form is saved to the database	Works as expected	Passed

Uploading required documents	Files are accepted	Works as expected	Passed
Admin updates scholarship status	Status changes on the user profile	Works as expecte	Passed
Sending email notifications	Email is received by user	Works as expected	Passed
API returns location	Location is displayed	Works as expected	Passed

Challenges and Solutions

During the development of the SK Gatid Scholarship and Grant WebBased Management System, we encountered several challenges. Here is how we addressed them:

Challenge	Issue	Solution
UI Responsiveness	Make sure the website is responsive and userfriendly on different devices (desktop, tablet, mobilephone).	Bootstrap 5.3.3 was used for responsive layout, images were adjusted, and CSS media queries were used.

Database Connection Errors (MySQL)	Occasional database connection error during development, due to initial configuration.	Adjusted database configuration settings, optimized queries, and added error handling to automatically reconnect.
Integration with Gmail (SMTP)	Problem sending emails (application updates, notifications) due to SMTP configuration.	The PHPMailer library was used to make it easier to configure SMTP settings. Proper authentication and encryption were set up to secure email sending.