UNIVERSITI TEKNOLOGI MARA

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

INFORMATION SCIENCE STUDIES

IMS566: ADVANCED WEB DESIGN DEVELOPMENT AND CONTENT MANAGEMENT

GROUP ASSIGNMENT

TITLE:

CLUB REGISTRATION MANAGEMENT SYSTEM (CLYRA)

PREPARED FOR:

FAIZAL HAINI BIN FADZIL

PREPARED BY:

NAME	STUDENT ID
AIMAN MUSTAQIM AMILIN BIN ZAINUDDIN	2023406952
NIK AMILA QISTINA BINTI MOHAMMAD SAUFI	2023213974
NURUL SHAMIMI ATHIRAH BINTI NORIZAL	2023680766

GROUP:

D1CDIM2624A

DATE OF SUBMISSION:

24th JULY 2025

Table of Contents

1.0 INTRODUCTION OF CLUB REGISTRATION MANAGEMENT SYSTEM (CLYRA)

1.1 BACKGROUND

Most educational institutions continue to allow club registration through printed A4 forms, and the painstaking process is still done manually by academic staff. There are some problems with this method such as losing the originality of the form before data entry, manually entering data is risky for human error, registering clubs took time, administration took time to check the authenticity of signature, human error took place again when approving the data. Therefore, as universities such as UiTM continue to emphasis to the students and staff about digital literacy and innovation, this paper-based club registration process is redundant and inefficient.

The creation of Clyra was born out of a need to modernize and simplify that process. "Clyra" comes from a combination of the word "club", together with the constellation "Lyra" which denotes harmony and unity. Likewise, Clyra exemplifies the goal of bringing club management together into a centralized, modern, and accessible system. As such, it will empower both students and administrators to more effectively manage the club registration process, club membership data management, and improve student involvement.

1.2 OBJECTIVES

Clyra has one primary purpose, to replace the manual recordkeeping system with a digital and organized platform to meet the needs of both students and administrators. More specifically:

- Students will be able to quickly and safely register for clubs online, using a simple online form.
- Administrators will have a systematic admin panel for approving member applications and managing clubs.
- Reduce the number of paper forms completed and the number of approvals that originate manually.
- Allow more students to engage in student clubs by providing a simpler method to access and register for clubs.
- Support the institution's current digital transformation initiatives by creating a real-life system solution.

2.0 PROJECT SCOPE

2.1 USER REGISTRATION OR LOGIN SYSTEM

The Clyra Club Registration System also features a safe and comfortable registration and log-in system so that access to the platform is limited among the evaluated users. Registrants are able to open an account through the process of supplying their level, email address, and academic program, together with a password, which is hashed through a secure reference to the password_hash() function in PHP to guarantee that unauthorized guests do not reach your account. The system allows the use of the role-based authentication, and the students and administrators have their specific permissions. Student users are able to log in and view available clubs or join them, whereas the administrators can access a special dashboard to keep the track of membership, as well as create reports. Session management will help in maintaining users log in as they browse as well as preventing unauthorized access.

2.2 CLUB MANAGEMENT FEATURES

The site includes all the functions of managing clubs, and the administrators are able to create, edit, and delete clubs. The details in each individual club listing are, the name of the club, the category (Sports, Academic, or Hobbies) and a description of what the club does. Before registering, students can visit available clubs, see the information associated with it. The system makes sure that the data of the clubs are sorted and can be accessed without much ado including the ability to filter the clubs according to category or search specific organizations. The administrators are allowed to change club information anytime and irrelevant information can be up to date to the students.

2.3 REGISTRATION SYSTEM

There is a user-friendly interface where students may see different clubs they may join. When a student makes an application in a club, it will fill in the system the name of the student, the club chosen and date of registration. The administrators have the ability to view these requests, accept them, or decline them and monitor the membership statuses (pending, approved or rejected). They are similarly recorded in the registration system and the old and present membership can be seen by the students and the administrators. This aspect guarantees transparency and can assist institutions to track club engagement in the long run.

2.4 ADMINISTRATIVE FUNCTIONS

The administrators get access to the powerful dashboard, which gives an overview about the activity of the system, such as the number of registered students, active clubs and pending requests. The dashboard contains user controls regarding the accounts of the students, editing the details of the clubs as well as processing registration requests. Also the administrators will be able to produce reports of club sign up, attendance and membership trends, and these reports can be exported to analyse them. Such administrative functions also simplify the work of managing clubs and commonly cut down on the manual effort of management and makes it more efficient on the part of an educational institution.

2.5 DATABASE STORAGE

The system is based on MySQL relational database that facilitates the storage and management of all important data such as student profiles, club and registration details. To keep the database data integrity, the database will consist of three primary tables, namely the students, clubs, and registrations tables, which are connected using the foreign key relationship. In case of deleting a student or a club, all the registrations that were associated with it are immediately deleted to avoid inconsistencies. Its database allows easy querying and this means that the membership lists, the student information as well as the club details will be retrieved within a very short time. This orderly process promotes the security of the stored data as well as ease of system operations.

3.0 DEVELOPMENT METHODOLOGY

3.1 METHOD USED

Clyra club registration system was created under the Agile system which follows the continuous improvement and step by step development process. Agile has been different, in that, unlike the traditional Waterfall method whereby a project is taken along a strictly-well-defined linear path, we managed to cut the project into smaller and more manageable units referred to as sprints. A sprint took 1-2 weeks and was dedicated to packaging a particular feature such as authentication of users, administration of the clubs, or reporting mechanisms. We had standard meetings such as a daily stand-up to monitor the progress, a sprint planning where we identified the tasks at hand, and a retrospective to redo our process. Such

flexibility allowed us to consider user feedback in the early stages of projects, adjust to the modifications, and make sure that the final product would match user needs, being of high quality at the same time. Agile was also iterative and this kind of approach allowed problems to be identified and solved early, so that severity of the problems are not compounded by the end of the development process.

3.2 SOFTWARE AND METHODOLOGIES USED

3.2.1 Visual Studio Code

Visual Studio Code is the primary code editor that we employed due to the high ratio of simplicity and the strength of the tool. Our PHP, HTML, CSS, and JavaScript were written and tested with the built-in syntax highlighting, code completion and debugging tools that allow the editor to be very easy to use. Valuable extensions like the PHP Intelephense to allow smarter PHP coding and the Live Server extension that allowed us instant previews helped us enormously in terms of productivity. The VS Code product has been preferred by us since it has a clean interface without too many contextual menu items and customization was also simple which enabled our team to work comfortably and have consistent coding standards across the project.

3.2.2 XAMPP Server

We selected XAMPP as our development environment since it offers ready to use and complete package that includes Apache web server, MySQL database and PHP. This complete package with an opportunity to launch a local testing server allowed us not to worry about the complex setups. The phpMyAdmin tool provided in it allowed us to manage our data base easily in the development phase. Through the XAMPP, we would be able to test the entire features at the local environment and resolve problems at an early stage before going on live. This method has saved us lots of time and made us make sure that our system would properly work once released.

3.2.3 Bootstrap 5 Framework

We used bootstrap 5 as our preferred frontend framework since it provided us with all the necessities to create a professional and responsive interface quickly. Its extensive library of ready-made elements such as navigation bars, cards, forms enabled us to design a uniform style to all

pages and time of development saved. Our layout automatically scaled to various screen sizes, using the mobile-first grid system, so we did not have to develop media queries that created intricate code. We adapted the default styles provided by Bootstrap according to our design principles without affecting the accessibility standards. The documentation and community of the framework were a trial to implement even in those members of the team who did not have much experience with frontend.

4.0 REQUIREMENTS ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Clyra has multiple important features designed with consideration for both end users (students) and admin users who oversee clubs. The following is a detailed description of the functional requirements and how these features will function within the system:

1. User Registration
   When new users sign up, they fill in their personal details, which provide their full name, student ID, email, phone number and a profile image. This develops a physical experience for the users and ensures complete data records.

2. Login/Logout
   The login system is role-based and lets us distinguish the difference between the user (student) and admin. We also implemented a secure session management system that helps maintain user states and user access features.

3. User Profile Management
   As users log in, they can manage their profiles. User can update their personal information, track their applications to clubs, and view announcements from clubs or events that are related.

4. Admin Data Management
   Admins are given full control of the backend system, allowing them to view pending club registration applications, approve or reject requests, update club details, and manage member lists. This ensures smooth and centralized oversight of club activities.

5. Data Storage in MySQL
   All user and club-related data is stored in a structured MySQL database. This ensures data integrity, supports efficient querying, and provides a foundation for scalability in the future.

4.2 NON-FUNCTIONAL REQUIREMENTS

Along with its primary functionalities, Clyra meets several crucial non-functional requirements that positively contribute to the overall performance, usability and reliability of the system. Below is a detailed breakdown of these non-functional requirements:

1. User Interface Responsiveness
   The system is designed with responsive design practices in HTML, CSS, and PHP. A significant consideration given the platform is student-focused and will be accessed on phones, tablets, and laptops.

2. Security
   Password hashing is used to protect user credentials, and sessions are used to help protect against unauthorized access. Admin functions are also limited and secured by verifying login credentials before accessing these functions. These measures protect the personal data of users and protect institutional data.

3. Email Verification
   Not included in the first release of the system; Email verification is designed into the system. Email verification provides added security and is useful for verifying new accounts and deterring spammers and duplicates.

4. Performance Expectation
   The system is built to accommodate increasing users without performance issues. Code optimizations and efficient database structures will reduce load times and enhance responsiveness, even with increased usage traffic.

5.0 SYSTEM DESIGN

5.1 SYSTEM FLOWCHART

The system flow has two user roles where the students and the admins. The action orders for the two groups start as follows. For students this flow starts with either register or login. As soon as users log in, they will be able to navigate a list of available clubs to apply to. After placing an application, users will then be able to check their application status that will be changed by the admin. For admins, this flow starts with logging into the admin dashboard. Admins will be able to view all applications submitted to join a club, they will be able to accept or reject each of the applications and manage the list of clubs and applicants.

A good flow will provide students a seamless way to apply to join the clubs and give admins a good system to manage.

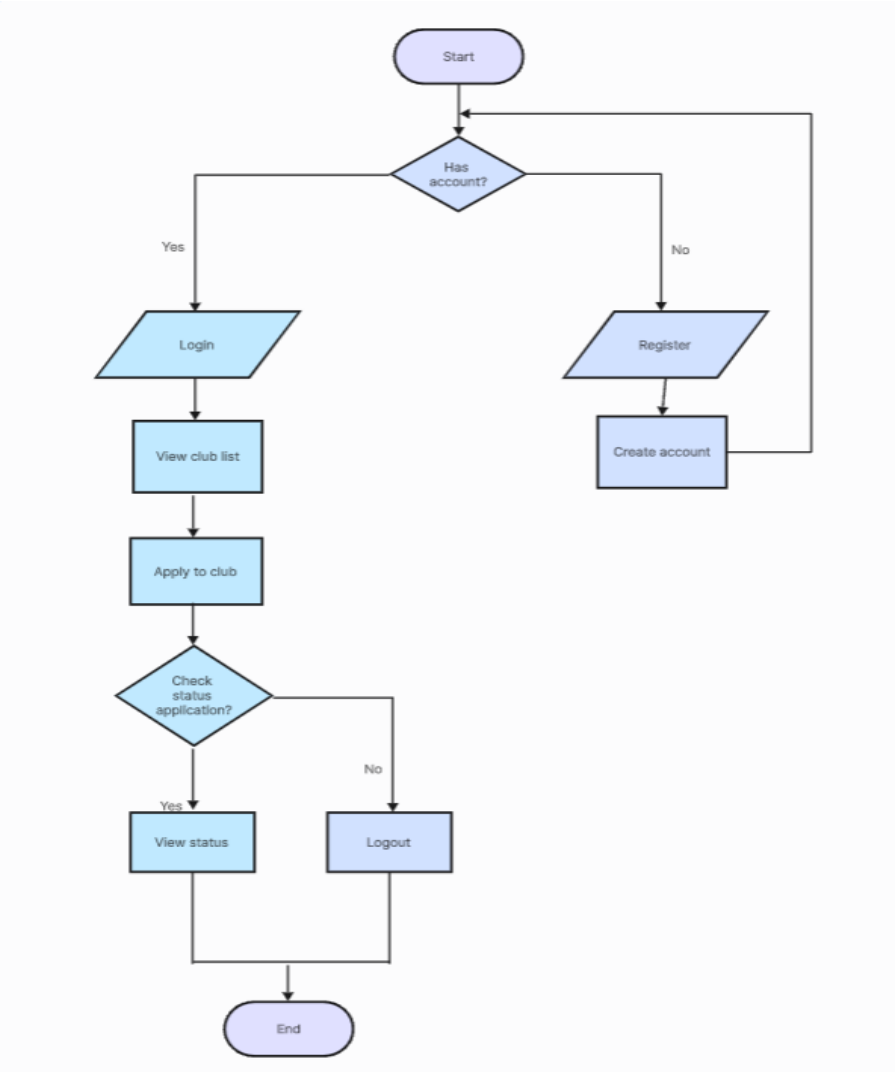

Figure 1: Flowchart of USERS

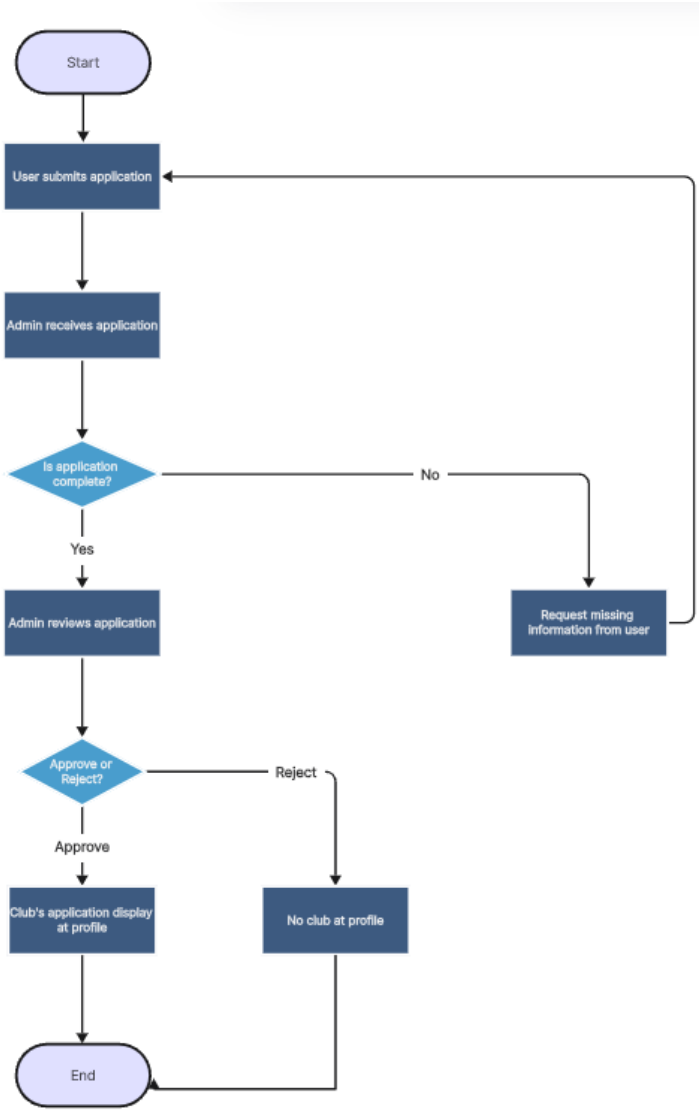

Figure 2: Flowchart of ADMIN

5.2 ENTITY RELATIONSHIP DIAGRAM (ERD)

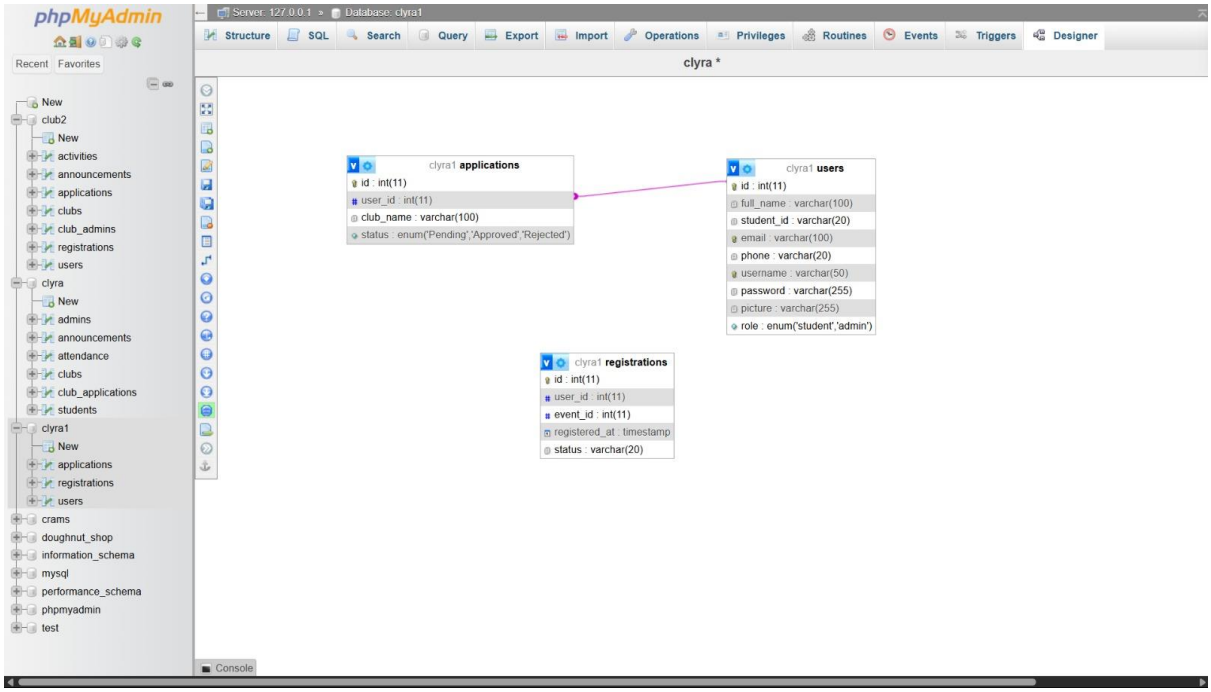

Figure 3: ERD of Clyra

5.3 DATABASE STRUCTURE EXAMPLE

CREATE TABLE `applications` (

`id` int(11) NOT NULL,

`user_id` int(11) NOT NULL,

`club_name` varchar(100) NOT NULL,

`status` enum('Pending','Approved','Rejected') DEFAULT 'Pending'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Dumping data for table `applications`

INSERT INTO `applications` (`id`, `user_id`, `club_name`, `status`) VALUES

(1, 2, 'AIS', 'Rejected'),

(2, 1, 'AIS', 'Approved'),

(3, 2, 'ENACTUS', 'Approved'),

(4, 2, 'MOBISM', 'Approved'),

```
(5, 7, 'HAC', 'Approved'),

(6, 6, 'ENACTUS', 'Approved'),

(7, 5, 'SISMA', 'Approved'),

(8, 5, 'PRIME MOVER', 'Rejected'),

(9, 5, 'DEBAT & PIDATO', 'Rejected'),

(10, 8, 'DEBAT & PIDATO', 'Approved'),

(11, 10, 'JPK TR', 'Approved'),

(12, 11, 'JPK TDM', 'Approved'),

(13, 11, 'SILAT CEKAK', 'Approved'),

(14, 12, 'JPK TR', 'Approved'),

(15, 13, 'SILAT CEKAK', 'Approved'),

(16, 15, 'JPK TAR', 'Approved'),

(17, 16, 'JPK THO', 'Approved'),

(18, 17, 'SILAT CEKAK', 'Approved'),

(19, 17, 'JPK NR', 'Approved'),

(20, 18, 'JPK DO', 'Approved'),

(21, 19, 'JPK THO', 'Approved'),

(22, 20, 'JPK THO', 'Approved'),

(23, 21, 'OMSA', 'Approved'),

(24, 1, 'MOBISM', 'Approved'),

(25, 24, 'SISMA', 'Approved');


-- --------------------------------------------------------

-- Table structure for table `registrations`

--

CREATE TABLE `registrations` (
```

```
 `id` int(11) NOT NULL,
`user_id` int(11) NOT NULL,
 `event_id` int(11) NOT NULL,
 `registered_at` timestamp NOT NULL DEFAULT current_timestamp(),
 `status` varchar(20) DEFAULT 'Pending'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

-

```
-- Dumping data for table `registrations`
INSERT INTO `registrations` (`id`, `user_id`, `event_id`, `registered_at`, `status`) VALUES
 (1, 1, 3, '2025-07-14 14:43:28', 'Approved'),
 (2, 2, 1, '2025-07-14 16:15:30', 'Approved'),
 (3, 7, 1, '2025-07-15 02:01:51', 'Approved'),
 (4, 6, 3, '2025-07-15 02:02:31', 'Approved'),
 (5, 5, 3, '2025-07-15 02:03:19', 'Approved'),
 (6, 5, 2, '2025-07-15 02:33:43', 'Rejected'),
 (7, 8, 1, '2025-07-15 03:58:31', 'Approved'),
 (8, 10, 2, '2025-07-16 07:42:23', 'Approved'),
 (9, 11, 3, '2025-07-16 07:45:47', 'Approved'),
 (10, 12, 2, '2025-07-16 07:48:01', 'Approved'),
 (11, 13, 2, '2025-07-16 07:50:02', 'Approved'),
 (12, 14, 1, '2025-07-16 07:52:12', 'Approved'),
 (13, 16, 3, '2025-07-16 07:56:57', 'Approved'),
 (14, 22, 1, '2025-07-16 08:10:20', 'Approved'),
```

(15, 17, 5, '2025-07-16 15:34:11', 'Approved'),

(16, 18, 4, '2025-07-16 15:35:04', 'Approved'),

(17, 18, 6, '2025-07-16 15:35:26', 'Approved'),

(18, 19, 6, '2025-07-16 15:35:57', 'Approved'),

(19, 20, 2, '2025-07-16 15:36:48', 'Rejected'),

(20, 21, 2, '2025-07-16 15:37:16', 'Rejected');

-- Table structure for table `users`

CREATE TABLE `users` (

`id` int(11) NOT NULL,

`full_name` varchar(100) NOT NULL,

`student_id` varchar(20) NOT NULL,

`email` varchar(100) NOT NULL,

`phone` varchar(20) NOT NULL,

`username` varchar(50) NOT NULL,

`password` varchar(255) NOT NULL,

`picture` varchar(255) DEFAULT NULL,

`role` enum('student','admin') DEFAULT 'student'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Dumping data for table `users`

INSERT INTO `users` (`id`, `full_name`, `student_id`, `email`, `phone`, `username`, `password`, `picture`, `role`) VALUES

(1, 'NIK AMILA QISTINA BINTI MOHAMMAD SAUFI', '2023213974', 'nikqistina27@gmail.com', '01110796813', 'Amila Qistina',

'$2y$10$p18ege56ONPI2qAUmKH4lOsafcilnYf/xCOKbVagYgwi5e9XRQMxa', 'pic_68754222c76a3.jpeg', 'student'),

(2, 'NOR AININ SOFIYA BINTI MOHAMMAD SAUFI', '2022216625', 'norsofiya1@gmail.com', '01110746813', 'Ainin Sofiya', '$2y$10$.Gh1.iNVtydabPrRInrIveATEUNIEAljJy1lL0J/lDYMwTK6YLiey', 'pic_68752fcb55642.jpeg', 'student'),

(4, 'NURUL SHAMIMI ATHIRAH BINTI NORIZAL', '2023680766', 'shamimi03@gmail.com', '01131083585', 'Shamimi Athirah', '$2y$10$NF2Wat3mNHIDcBzmuSGcA.lwaix2Uhc7Hat8V5giqyVRxYYG7HmnS', 'pic_6875423eeb4cf.jpg', 'student'),

(5, 'FARAH HASINA BINTI ZAIDI', '2023298104', 'farahhasina@gmail.com', '0107627694', 'Farah Hasina', '$2y$10$S/H8OQ5DITR4O1Nxfl9uEOBKcNmLaStTpLeUjf2OVZy7x7qWmmOIC', 'pic_6875b5aecb0f9.jpg', 'student'),

(6, 'NUR ALIS AQILAH BINTI ZAHARI', '2023492036', 'alis03@gmail.com', '01129204138', 'Alis Aqilah', '$2y$10$5uRed.X53Fsml0thXTfjGO6Y3EcBiXfUCvJr2nxSORyGVxQ9VY6By', 'pic_6875b61b50c96.jpg', 'student'),

(7, 'NURULNISA IMANINA BINTI ABDUL MUHAIMIN', '2023225656', 'nisa07@gmail.com', '0179629780', 'Nisa', '$2y$10$SHzXUh60F05C.lb3ahG7Hum09tkCVRtZICl5Shj9TDmm3U4Rm5EBe', 'pic_6875b66d968e8.jpg', 'student'),

(8, 'MUHAMMAD AFIQ BIN MD HUSIN', '2023239018', 'afiq00@gmail.com', '0185700465', 'Afiq ', '$2y$10$3wPI.9zCSXivkVIUwJXp9.dKlCUB7zNsFF6FZQonm0S1dLW0TT/PG', 'pic_6875d16bc3302.jpeg', 'student'),

(9, 'MUHAMAD AMAR BIN ZAINAL', '2023001567', 'amarzainal01@gmail.com', '0174567892', 'Amar', '$2y$10$1EndBJDAsryTKTtplCkBl.cSxNWwPB52E0nDEXTuSMMh3XALWsupG', 'pic_6877623a26a59.webp', 'student'),

(10, 'AHMAD FARIS BIN RAMLI', '2023002385', 'ahmadfaris@gmail.com', '0137853421', 'Ahmad Faris',

'$2y$10$zNmbCA3971Vajrd6E1fTj.i2S7TR2fnLvYF94ZlMWmst93Vbh9fVi', 'pic_68776248bf8d3.png', 'student'),

(11, 'HAFIZUDDIN BIN ZULKIFLI', '2023003972', 'hafizzulkifli@gmail.com', '0112683947', 'Hafizuddin', '$2y$10$T7NKHnVUghe4AMwTze4Es.BtbSmoUOLrBEFRFk0uD0ACQ20VCG9ba', 'pic_6877627128397.jpg', 'student'),

(12, 'MUHAMMAD IRFAN BIN RAHMAT', '2024001283', 'irfanrahmat24@gmail.com', '0129372651', 'Irfan', '$2y$10$jR1Jr.GkF2yFlMb1ScN7nOqO0sjll6xsevX.lMFs9Y5o9VTsbMXdK', 'pic_687763024d874.jpg', 'student'),

(13, 'DANIAL HAKIM BIN ABDULLAH', '2024002448', 'danialhakim@gmail.com', '0105281930', 'Danial Hakim', '$2y$10$GrimNjzrHoQnIDNVDKYr2./8IDyd7ZFJ0kis4R6iAWMhqLKg6ivUS', 'pic_68776398169f4.jpg', 'student'),

(14, 'MOHD FAIZ BIN HASSAN', '2024003069', 'faiz92@gmail.com', '0164728610', 'Faiz', '$2y$10$/uIM8s.lpzXZ2.K3pF8sS.ttfO2wDr5fyPZRUOZ2uwwVifwQEb666', 'pic_687763a5e8d99.jpg', 'student'),

(15, 'SITI ZULAIKHA BINTI SYED ZAMRI', '2023001883', 'zulaikha12@gmail.com', '0147652193', 'Siti Zulaikha', '$2y$10$al8Dk5MqoPHNpJMNoswwj.irTFZ3CxNG.wZmLF9ZEoB1TOjwAySTO', 'pic_687764601544c.jpg', 'student'),

(16, 'NUR AISYAH BINTI ZAKARIA', '2023002574', 'aisyahzakaria01@gmail.com', '0183856210', 'Aisyah', '$2y$10$l9x5MxY9cP1yMLjW.kuECOUErPiCODgEPGW1LuGlul.2gMh.qn5CO', 'pic_687764720e223.jpg', 'student'),

(17, 'NURUL IMAN BIN HAMDAN', '2023003912', 'imanhamdan@gmail.com', '0197341280', 'Nurul Iman', '$2y$10$pS25BUeGxGkNe/h01ysNa.A/QyWlSw/3sbH5i8caXuPVNJyFe1Iyq', 'pic_687764835b28a.jpg', 'student'),

(18, 'AINA SOFEA BINTI JAMALUDDIN', '2023004768', 'ainasofea@gmail.com', '0115629402', 'Aina Sofea',

'$2y$10$OVUOfbDiE8zWXf1i/DS2h.Yfu/p//mUqahBBqYpMNU8zwKYdZGkgy', 'pic_687764d5f2db8.jpg', 'student'),

(19, 'BALQIS NAJIHA BINTI ROSLAN', '2024001029', 'balqisnajihah07@gmail.com', '0176497238', 'Balqis Najihah', '$2y$10$pLqJc4MlrfZa4YXiCRgmPO.Ut/rZ/Fg49DTW1iT9Y7bDLUuUzWQB6', 'pic_6877661b31127.jpg', 'student'),

(20, 'NURSYAFIQAH BINTI MOHD RIZAL', '2024002184', 'syafiqahrizal@gmail.com', '0139087356', 'Nursyafiqah', '$2y$10$Ihn1T/SbUOtI6EqwWrOYbeVZPt2ZNWIezuJKTA4yudYUg662XbquC', 'pic_6877658109bdb.png', 'student'),

(21, 'WAN NUR AMIRAH BINTI FAIZAL', '2024003047', 'wanamirah05@gmail.com', '0128345701', 'Nur Amirah', '$2y$10$gzlRzYSQPFLDMMqM76NHM.Rm9lgI5Ft6dsDiNpUN6zAG95i/df9qW', 'pic_6877663455128.jpg', 'student'),

(22, 'NABILA SYAHIRAH BINTI AHMAD ', '2024003650', 'nabila06@gmail.com', '0162956743', 'Nabila Syahirah', '$2y$10$eltcP7xsJey8aB5IV9PHJOYzz/OT2Ew.glJTlFUEDCdwAjgA9o1.e', 'pic_6877667dd6bbf.jpg', 'student'),

(23, 'FARAH DINA BINTI SHAHRUL', '2024004291', 'farahdina@gmail.com', '0187624910', 'Farah Dina', '$2y$10$MfjifVDuo1DN.vKmwuwXHe16UJ44imkkogz9NvD81FGzpIw6ygmda', 'pic_6877668da1d76.jpg', 'student'),

(24, 'Nurul Izzaati binti Yusoff', '2023299688', 'izzati@gmail.com', '01110786512', 'Izzati', '$2y$10$8dU/ufnN80eMCjUqmYYCc.sNqS.wsYuYKFzehmm/FtaD25.HmqoIK', 'pic_6881c5e8db7b9.jpeg', 'student');

-- Indexes for dumped tables

-- Indexes for table `applications`

--

ALTER TABLE `applications`

ADD PRIMARY KEY (`id`),

```
ADD KEY `user_id` (`user_id`);

-- Indexes for table `registrations`

--

ALTER TABLE `registrations`

ADD PRIMARY KEY (`id`),

ADD KEY `user_id` (`user_id`),

ADD KEY `event_id` (`event_id`);


-- Indexes for table `users`

ALTER TABLE `users`

ADD PRIMARY KEY (`id`),

 ADD UNIQUE KEY `email` (`email`),

ADD UNIQUE KEY `username` (`username`);

-- AUTO_INCREMENT for dumped tables

-- AUTO_INCREMENT for table `applications`

--

ALTER TABLE `applications`

MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=26;

--

-- AUTO_INCREMENT for table `registrations`

--

ALTER TABLE `registrations`

MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;

-- AUTO_INCREMENT for table `users`
```

```
--

ALTER TABLE `users`

 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=25;

-- Constraints for dumped tables

-- Constraints for table `applications`

--

ALTER TABLE `applications`

ADD CONSTRAINT `applications_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE;

COMMIT;
```

6.0 USER INTERFACE

6.1 HOMEPAGE AND AUTHENTICATION

The responsive design of the authentication system provides a simple interface of card-based login (login.php). It has the right HTML5 validation of the email and password fields (required fields) in the login form and appropriate error messages are shown when a wrong password is used ("Incorrect password" or "Email not found", which are shown in Bootstrap alert elements). The colour scheme is taken essentially to be Bootstrap purple colour scheme where the lengths are being kept white just to keep things consistent. There is a visible link of registration shown below the form to make it convenient to the user. Students, just by performing the authentication process over MySQL are automatically hosted into the dashboard without the need of another authentication process on the dashboard through verification of session.

6.2 STUDENT DASHBOARD

When students login this website, they are presented with an easy-to-use interface (student/index.php) in the form of a responsive grid (row/col system) that displays the clubs that are available. The list of clubs is generated in a form of Bootstrap cards with similar styling, each card has a header with a name of the club (card-header), paragraph with a description (card-body), and a tag with a category (card-badge). The consistency of the Register-Now button in every footer of the cards is adopted and there exists the existence of clear calls-to-action. It has a three-tab profile menu whose navigation is achieved easily with the use of active state styling of Bootstrap. The interface is also i-phone responsive by following the grid system provided by bootstrap but also provides a sufficient amount of white space and font consistency to be readable.

6.3 ADMIN DASHBOARD

On the administration panel (admin/index.php) important statistics (numbers of clubs, students, registrations) are shown using separate cards that are visually different and labelled with icons and use big font size to make them optimally scannable. The management button has cog icons all over each metric card, whereby the concept of the control is maintained by the admins. Its layout is composed of the flexbox utilities provided by Bootstrap (d-flex, justify-content-between) in order to achieve a balanced header with logout capabilities. The dashboard uses font awesome to design visual hierarchy and have the same purple/white system-wide colour scheme, adding minimal shadows and hover effects to improve navigation to interface elements.

6.4 CLUB MANAGEMENT

Admin club management (admin/clubs.php) display a fully functional table interface include column that being able to be sorted and actions (edit, delete). The table employs Bootstrap table-hover class to highlight the rows and added truncated descriptions enclosed with ellipse to consume the minimal space. There is a large bright button with a protruding plus sign in the header, which is positioned in accordance with the colour scheme of the actions of the system. To avoid accidental deletes, JavaScript confirmation dialogs piece is initiated. Its responsive table wrapper helps it to display properly on the mobile device and components of badge gives a visual classification of the clubs. The edit/delete buttons take outline types with icons as labels so as to retain functionality without cluttering the other cells in the action column.

7.0 SYSTEM TESTING

Test the system to verify the website and the registration function prior to releasing it. System testing allows for verification of bugs or bugs threatening security, and low performance that impact user experience. Testing was broken into three main areas which is functional testing, security testing, and performance testing. Each area of testing has a specified aim to verify the system meets the project requirements and provide a good user experience for students and administrators.

7.1 FUNCTIONAL TESTING

Functional testing was completed to verify that every important functional aspect of the club registration website, operates correctly based on the system requirements. Functional testing oriented itself around verifying that the expected output occurs for given inputs and triggers for all user interface tasks that needed testing. Essentially, the objective of functional testing was to ensure that students and admins could interface with the application without functional issues encountered with the interaction system as it pertained to the application.

7.1.1 Login Functionality

- The login functionality was tested for both user types, students and administrator users. Valid credentials successfully directed both user types to their dashboards, and invalid usernames or passwords resulted in the appropriate error messages. The login functionality also establishes role-based accessibility, such as organizing content-instruction options for additional administrator user tools that are not offered to normal users.

7.1.2 CRUD Operations

CRUD (Create, Read, Update, Delete) operations were tested with two user roles, students (users) and admin, who both interact with different data elements in the system.

- Admin

  CRUD testing was performed on the club management aspects of the project. Admins can create a new club record by entering the club's name, category, and description. Admins can read and view existing clubs. Admins could update the club option if change was needed (for example, if a club name was misspelled or a description needed editing) and delete a club record if it was no longer active. These operations were tested with the approach of observe how the code was designed to have the system sync to the database and examine the database to see if the delete action was reflected correctly with no broken links or residual data.

- Students

  CRUD operations were tested on club application processes. Students could create applications to belong to clubs, which had a database pending status assigned to them. After they created an application for a club, they could read the status of their application in their dashboard. In some versions students were provided with updates or cancellations for pending applications prior to being viewed by an admin. Again, regarding deletion an application was deleted from both the student's view, and the admin's view.

The operations were then tested against multiple scenarios to ensure that the operations were reliably carried out as functions and that the database records were updated correctly and without error in relation to the user actions. Additionally, the interface was also tested to confirm that the appropriate messages were shown for the user after an action for persistent or deletion of records.

7.1.3 Search Functionality

The search bar has been tested so students can search clubs by title or category. The system frowns filtered results based on the keywords included. Therefore, it is an improved browsing function because it becomes more user-friendly, particularly when there are many clubs.

7.2 SECURITY TESTING

In order to establish that the club registration website is secured against possible vulnerabilities and unauthorized access, security testing was undertaken. This kind of testing is imperative in preserving the confidentiality, integrity, and availability of the system, when dealing with sensitive information such as user credentials and application records within a club membership registration application.

7.2.1 SQL Injection Defense

A major focus of our security testing was SQL injection, where an attacker attempts to interact with the application's database by including SQL commands in the user input. The way to test this is to use input such as '; DROP TABLE users; -- in the login form and other text fields. In this particular test, an attacker is trying to terminate the original SQL command and input a destructed command that would wipe out the entire users table, assuming that the system was vulnerable to command manipulation. The website blocked each attempt to manipulate the SQL command using prepared statements and input sanitization techniques. Thus no data was compromised, and

the state of the database was kept intact under all conditions of the tests.

7.2.2 Form Validation

All user-facing forms (registration, log-in, club application, etc.) were reviewed to confirm correct form input validation. Each user-facing form was assessed to confirm that input was validated for empty required fields and correct input type. For example, the email field required an email format (e.g, Ahmad@gmail.com) and password fields had minimum length rules. If data was submitted incorrectly or incompletely, the interface displayed clear plain English that informed the user how to replace their incorrect input with appropriate input. Good form validation enhances usability and also prevents faulty or harmful data from entering the system.

The security testing showed that the system is configured to best practices regarding basic vulnerabilities and secure interaction for students and administrators. We recommend regular security reviews as the system scales to maintain security going forward.

7.3 PERFORMANCE TESTING

Performance testing was performed to capture the interactions of the user with the club registration website and assess how responsive the site will remain in different environments and devices. Performance testing is geared to give users fast, smooth and reliable performance which is especially important in web-based interfaces that deal with real-time data submissions and user sessions.

7.3.1 System Responsiveness

Responsiveness of the system was tested in terms of response time for login, browsing clubs, submitting applications, and loading all other pages. Various actions were given a time score to see if they completed in an acceptable range, typically elapsed time under 2 seconds. All response times from the performance testing indicated that

the website loaded promptly and various operations, such as applying to a club and checking application status were processed without observable latency. It clearly indicates that the website has been optimally defined and database queries are being processed in a timely fashion.

7.3.2 Cross-Browser Compatibility

The system was also tested on different browsers that have large community usages: Google Chrome, Microsoft Edge, and Mozilla Firefox. The same actions were done for each browser (e.g., login, search, CRUD operations) to verify layout explained the visual layout was consistent, styling was accurately rendered, related functionality worked. The website did have the same caliber of appearance and behavior for all browsers tested, without layout performance issues, or broken functionality. This is beneficial for the wider accessibility users can have with independent browser choice.

In conclusion, functional performance testing demonstrated reasonable results; the system displayed both speed, and reliability. It is a recommendation that systematic performance monitoring and optimization be performed on a regular basis while the system is transitioned from the pilot to general use to improve system accessibility as more users are continuing to access the system, new features are added and being utilized.

8.0 IMPLEMENTATION SCHEDULE

Week	Start Date	End Date	Task	Key Deliverables
1-2	13/05/2025	26/05/2025	Problem Research & Requirement Gathering	- Defined project objectives - User needs assessment - Functional requirements document
3-4	27/05/2025	09/06/2025	System & Database Design	- ER diagrams - Database schema - UI wireframes - System architecture plan
5-6	10/06/2025	23/06/2025	Backend & Frontend Development	- Working authentication system - Admin dashboard - Student interface - Club management features
7-8	24/06/2025	7/07/2025	System Testing & Bug Fixes	- Unit test results - User acceptance testing - Bug reports and fixes - Security validation
9	08/07/2025	14/07/2025	Final Documentation	- Technical documentation - User manuals - Deployment guide
10	15/07/2025	22/07/2025	Project Presentation	- Live demo - Presentation slides - Q&A preparation

9.0 CONCLUSION

Clyra is a positive development for advancements made in modernizing club registrations at an academic institution level. Empowering the once exclusively paper-based system into a centralized, web-based system like Clyra, will solve many ongoing issues such as inefficient practices, inaccuracies with data, and time lost on administrative work. Clyra conveys a range of features which include, but is not limited to user registration, a secure login/logout process, a detailed user profile with personalized editing, a comprehensive dashboard for administrators to manage their clubs and student's applications. These features will move both students and administrators through the system smoothly and efficiently.

Additionally, to the functional aspects, Clyra includes non-functional aspects such as responsive design, basic security, and performance enhancement, allowing the system to be accessible, reliable, and useful when accessed through different devices and use cases. The present version of Clyra has already been adapted to the achievable goals of the project, but future improvements could be accessible that could enhance it even further, including email verification, a reward system, push notifications for real-time access, and analytics dashboards to support decision making.

Overall, Clyra is a vital resource for simplifying club management and encouraging student involvement. It is not only an easy way to manage administrative tasks, but also a great way to inspire students to engage in campus life. As post-secondary institutions continue to embrace digital processes, Clyra is a useful and meaningful platform that utilizes technology toward the goal of connected and efficient university life.