1.Registration Process

The registration process begins when a student enters their admission details into the system. The system performs both client-side and server-side validation to ensure all required fields, such as name, email, and admission ID, are filled correctly. If any field is missing or incorrectly formatted, the system halts the process and displays an error message prompting the student to correct the input. If the validation is successful, the system forwards the student's details to the Register API via a POST /api/register request. The Register API first checks the database to determine whether the student already exists based on their email or admission ID. If an existing record is found, an error is returned, informing the student that an account with the provided details already exists.

If the student does not already exist in the database, the Register API proceeds with the registration process by inserting the student's details into the Students table. At this stage, the API also generates a verification token (such as a UUID or JWT) and stores it in the StudentVerification table. The token is linked to the student's record, marked as unverified, and assigned an expiration time (e.g., 24 hours). After successfully generating the verification token, the Register API triggers the Email API, which sends a verification email to the student. This email contains a verification link (e.g., https://academy.com/verify?token=<token>) and the student's login credentials. If the email fails to send, the admin is notified, and the student remains unverified. However, if the email is sent successfully, the student receives the verification instructions and can proceed to verify their account.

The final step depends on whether the student completes the verification process before the token expires. If the student clicks the verification link within the allowed timeframe, the system updates the StudentVerification table by marking the student as verified, clearing or nullifying the token and expiration time, and allowing the student to log in. However, if the student fails to verify their account before the token expires, the system automatically removes unverified records using a scheduled API (e.g., a cron job). This scheduled cleanup process deletes records from both the Students and StudentVerification tables where verification has not been completed. As a result, the student must wait 30 minutes before re-registering, at which point they can attempt the process again with a fresh token and admission details. This ensures database integrity while allowing unverified students to reattempt registration if needed.

2.Login process

The login process begins when a student navigates to the login page and enters their Student ID and Password. The system first performs client-side validation to ensure that the input fields are not empty and that the Student ID follows the correct format (e.g., STU/12345678/25). If the validation fails, the system prevents further processing and displays an "Invalid input" message. However, if the input is valid, the system sends a POST request to the Login API (/api/login) with the provided credentials.

Upon receiving the request, the Login API first authenticates the student by querying the Students table in the database to check if the provided credentials match an existing record. If no match is found, the API returns a 401 Unauthorized error with an "Invalid credentials" message, preventing access. However, if the credentials are correct, the system proceeds to check verification status. The Login API queries the StudentVerification table to determine whether the student has verified their account. If the student is not verified, the API stops the login process and returns an error message instructing the student to complete the verification process before logging in. If the student is verified, the login process proceeds to session management.

For session handling, the system first checks the Sessions Table to determine whether the student already has an active session. If an active session exists, the system deletes the existing session to enforce a one active session per user rule. This prevents multiple concurrent logins using the same Student ID. After clearing any previous session (if applicable), the system creates a new session, generating a unique sessionId and setting an expiration time (e.g., 24 hours from login). The session details are then stored in the Sessions Table. Finally, the system confirms a successful login, allowing the student to access the platform.

Winnas Academy System Flow – Detailed Overview

The Winnas Academy platform follows a structured process to ensure a smooth user experience, from login to course enrollment, payments, and access to online classes.

If a student forgets their password, they can initiate the recovery process by clicking the "Forgot Password" link and entering their Student ID. The system then queries the database to check if the Student ID is registered. If found, the system triggers the email service to send the login credentials to the student's registered email. However, if the ID is not found, an error message is displayed informing the student that the provided Student ID is not registered.

Once logged in, the student is directed to the dashboard, which provides a real-time summary of their profile, fee status, and timetable. The system fetches this data from the database and updates the dashboard dynamically. If a student wishes to enroll in a course, they can select a course from the available options. The system then registers the course under the student's ID in the database. If enrollment is successful, a confirmation popup appears, and the page reloads to update the fee summary. If enrollment fails, an error message is displayed.

For fee payments, students can use PayPal, ensuring that payments are only made for enrolled courses without exceeding the total course amount. Once a student initiates a payment request, the system validates the request and sends it to the Payment Gateway for processing. If the payment is successful, the system updates the payment record in the database and triggers the email service to send a transaction receipt to the student's registered email. However, if the payment fails, the system notifies the student with an error message.

Under the Online Services menu, students can access Online Classes to view their scheduled courses. The system fetches the student's enrolled courses from the database and checks the payment status. If a course has been fully paid for, the class link is displayed. However, if payment is incomplete, the course appears without a class link, ensuring that only fully paid students can access online lessons. If a student

has no enrolled courses, the system displays a message indicating that no courses are available.

Finally, when a student chooses to log out, the system removes their active session from the database and clears their session ID from local storage. The student is then automatically redirected to the login page. Throughout all these interactions, data is fetched and updated in real-time using APIs, ensuring that students always access the latest information.