Here’s a **complete website workflow** for the **Witquest Castle School** website, incorporating all your requirements, using **PHP**, **MySQL**, and **Bootstrap**.

**Complete Website Workflow**

**1. Homepage**

**Frontend (Bootstrap):**

* **Hero Section**:
  + Carousel with school images.
  + Call-to-action button: *"Learn More About Us"*.
* **Quick Links**:
  + Links to Results Portal, Admissions Portal, Activities, and Contact Us.
* **Activities Highlights**:
  + Cards showing different activities (e.g., sports, arts, clubs).
* **Facilities Section**:
  + Display key facilities like sports grounds, library, classrooms using Bootstrap grid.

**Backend (PHP & MySQL):**

* **Dynamic Content**:
  + Display school announcements, upcoming events, and notices from the database.

**2. About Us Page**

**Frontend (Bootstrap):**

* **Mission & Vision**:
  + Short description of the school’s mission and values.
* **Leadership Team**:
  + Display profiles of key personnel (Principal, Head of Departments, etc.) in Bootstrap cards.
* **Achievements**:
  + Show notable achievements (academic, sports, etc.).

**Backend (PHP & MySQL):**

* **Dynamic Content**:
  + Store school history, mission, leadership, and achievements in the database for dynamic rendering.

**3. Academics Page**

**Frontend (Bootstrap):**

* **Tab Navigation**:
  + Tabs to show curriculum for Playgroup, Primary, and Junior Secondary.
* **Curriculum Accordion**:
  + Show curriculum breakdown in an accordion format.

**Backend (PHP & MySQL):**

* **Curriculum Management**:
  + Store curriculum information in the database and fetch dynamically to display it.

**4. Results Portal**

**Frontend (Bootstrap):**

* **Login Prompt**:
  + If a parent hasn’t signed up, they must sign up first.
  + After registration, a login page is shown with fields for phone number and password.
  + Upon successful login, the parent is redirected to the results page.
* **Results View**:
  + After logging in, the parent can view their child’s results:
    - Subject Marks, Grades, Total Marks, and Rank.
    - "Download Report Card" button to download results in PDF.

**Backend (PHP & MySQL):**

* **User Authentication**:
  + On login, the system checks if the phone number exists in the users table. If found, it matches the password using bcrypt hash.
* **Password Reset**:
  + If a parent forgets their password, they can request a reset link, which sends a secure reset token via email or SMS.
  + The link allows the parent to set a new password.
* **Results Management (Admin)**:
  + Admins can enter student marks (subject marks, grades, and ranks) for each student by class.
  + **Admin Dashboard** allows managing student results across Playgroup, Primary, and Junior Secondary.

**Database:**

* **Tables**:
  + students: stores student names, roll numbers, and grades.
  + results: stores subject marks, total marks, grades, and ranks.
  + users: stores parent details (name, phone number, password).

**5. Admissions Portal**

**Frontend (Bootstrap):**

* **Parent View**:
  + **Payment Verification**: Parents can view the payment status (e.g., Paid/Not Paid).
  + If fees are paid, a "Download Admission Form" button is displayed.
  + If not paid, the message *"Please pay the school fees first"* is shown.
* **Admin View**:
  + Admins can verify payment by checking the bank receipt or MPESA code, then mark it as “Paid” in the system.
  + After payment is marked as verified, the parent is allowed to download the admission form.

**Backend (PHP & MySQL):**

* **Payment Verification**:
  + When a parent clicks on "Get Admission Form", the system checks if the parent’s payment code (receipt number/MPESA code) exists in the payments table. If it exists and is marked as paid, the parent can download the form.
* **Admin Panel for Payments**:
  + Admins enter payment information into the system, including the receipt number or MPESA code and the associated phone number.
  + Admin marks payments as confirmed after verifying from the finance department.

**Database:**

* **Tables**:
  + payments: stores payment details (receipt number, MPESA code, phone number, status).

**6. Activities Page**

**Frontend (Bootstrap):**

* **Activities Overview**:
  + Use cards or grid layout to display different school activities (e.g., sports, clubs, arts).
  + Each activity card links to more details.

**Backend (PHP & MySQL):**

* **Activity Management**:
  + Store activities data (name, description, schedule) in the database.
  + Admin can add, edit, or delete activities.

**7. Facilities Page**

**Frontend (Bootstrap):**

* **Facilities Overview**:
  + Grid or list layout to show key school facilities like libraries, classrooms, sports grounds, and labs.
  + Optionally, include a 360-degree tour or video.

**Backend (PHP & MySQL):**

* **Facilities Management**:
  + Admin can add/edit facilities in the database and display them dynamically.

**8. Contact Us Page**

**Frontend (Bootstrap):**

* **Contact Form**:
  + Name, email, phone, and message fields.
  + Submit button to send the message to the admin.
* **Google Map Embed**:
  + Display the school’s location on Google Maps.

**Backend (PHP & MySQL):**

* **Message Handling**:
  + Admin can view messages sent through the contact form and respond via email.

**9. Admin Dashboard**

**Frontend (Bootstrap):**

* **User Management**:
  + Admins can manage all users, including resetting passwords, updating usernames, and activating/deactivating accounts.
  + Admin dashboard includes sections for managing:
    - Results: Add/edit student results.
    - Payments: Verify payment status.
    - Activities: Add/edit school activities.
    - Facilities: Manage facilities information.
* **Password Reset**:
  + Admin can reset a user's password and send a reset link.

**Backend (PHP & MySQL):**

* **CRUD for Users**:
  + Admin can add, edit, delete, and view all users (parents, staff).
* **Password Reset**:
  + Admin can generate a password reset token for a user (parent) if they forget their password.
  + Admin can send a reset link to the user for resetting their password.
* **Session Management**:
  + When a user logs in, the session is updated. If they change their password or username, the session is refreshed with the updated information.
* **Database Tables**:
  + users: store usernames, passwords, roles (admin/parent), and statuses (active/inactive).
  + activity\_logs: stores admin actions for auditing purposes.

**10. Security Considerations**

1. **Password Encryption**:
   * Use **bcrypt** for password hashing.
2. **Token-Based Reset**:
   * Use unique, time-limited tokens for password resets. Token is stored with an expiration time in the database.
3. **Session Management**:
   * After login, ensure session data reflects the latest updates (e.g., new username/password).
4. **SQL Injection Protection**:
   * Use prepared statements with MySQL queries to prevent SQL injection.
5. **Input Validation**:
   * Validate user inputs (e.g., phone numbers, emails, etc.) before inserting them into the database.

**User Flow**

1. **Visitor Flow**:
   * Visitors can browse the homepage, About Us, Academics, Activities, and Facilities pages.
   * To access results or the admission form, the user needs to register (if not already registered).
2. **Sign-Up and Login Flow**:
   * If a parent hasn’t signed up yet, they are prompted to register first (name, phone number, password).
   * After registration, they must log in with their phone number and password.
   * If they forget their password, they can reset it via a link sent to their phone number or email.
3. **Results Portal**:
   * After login, parents can view their child's results. If needed, they can download a report card.
4. **Admissions Portal**:
   * Parents can check if fees are paid (verified by the admin). If fees are paid, they can download the admission form.
5. **Admin Dashboard**:
   * Admins can manage users (reset passwords, change usernames) and handle results, payments, activities, and facilities.

**Next Steps for Implementation**

1. **User Registration**: Implement user registration and login functionality.
2. **Password Reset**: Implement the token-based password reset system.
3. **Admin Panel**: Set up the admin dashboard to manage results, payments, activities, and users.
4. **Security Measures**: Apply necessary security measures such as password hashing, token expiration, and SQL injection prevention.