==>Functional Requirement of the Hostel Management System Project:

1. It should allow admins to add hostel students by creating their accounts by adding information like name, hostel, course, email, branch, mobile no., room no. and managing their fees.

2. It should allow admin to add hostel staff by making their accounts, editing their information and managing their pay.

3. It should allow the admins to access profiles and view information about the staff and the students.

4. It should allow admins to add and remove other admins and manage them.

5. It should allow students, staff and admins to view and edit their information and profile.

6. It should allow students to file complaints while having the option to state their urgency.

7. It should allow the students to give suggestions to the hostels and management while also categorising them.

8. It should allow the students and the admin to check the due states of the hostel rent and any other corresponding charges.

9. It should allow the admin to renew the students' registration every year quickly.

10. It should allow push notifications onto the site and students whenever their due date is close or has passed by.

11. It should have the functionality to verify users based on their email and by sending an OTP.

12. It should represent all the hostels and necessary links to other sites.

==>The Non-Functional Requirements of the Hostel Management System are:

1. Scalability: The system should be usable for many students and multiple hostels.

2. Reusability: The system should be easily be replicated and reused for other campuses or Colleges.

3. Security: The system should be secure and prevent tampering or messing with data without permission.

4. Portability: The system should work with different devices, operating systems, versions of libraries and software.

5. Maintainability: The system should be easily maintainable with the proper backing of database management software and should maintain integrity with changes in users and other software dependencies.

6. Efficiency: The system should work on decent hardware, use fewer resources, be snappy, and adhere to time constraints.

7. Readable & Compahendable: The code should be neat, clean and made easy to read and edit by other developers through comments.

8. Easy-to-use: The software should be reasonably easy to use with interactive UI and rigid structures.

9. Flexibility: provisions to use different datasets and database management systems if needed.

10.Testability: The software can be tested with data before being deployed to check if the verification and security features are working as intended.