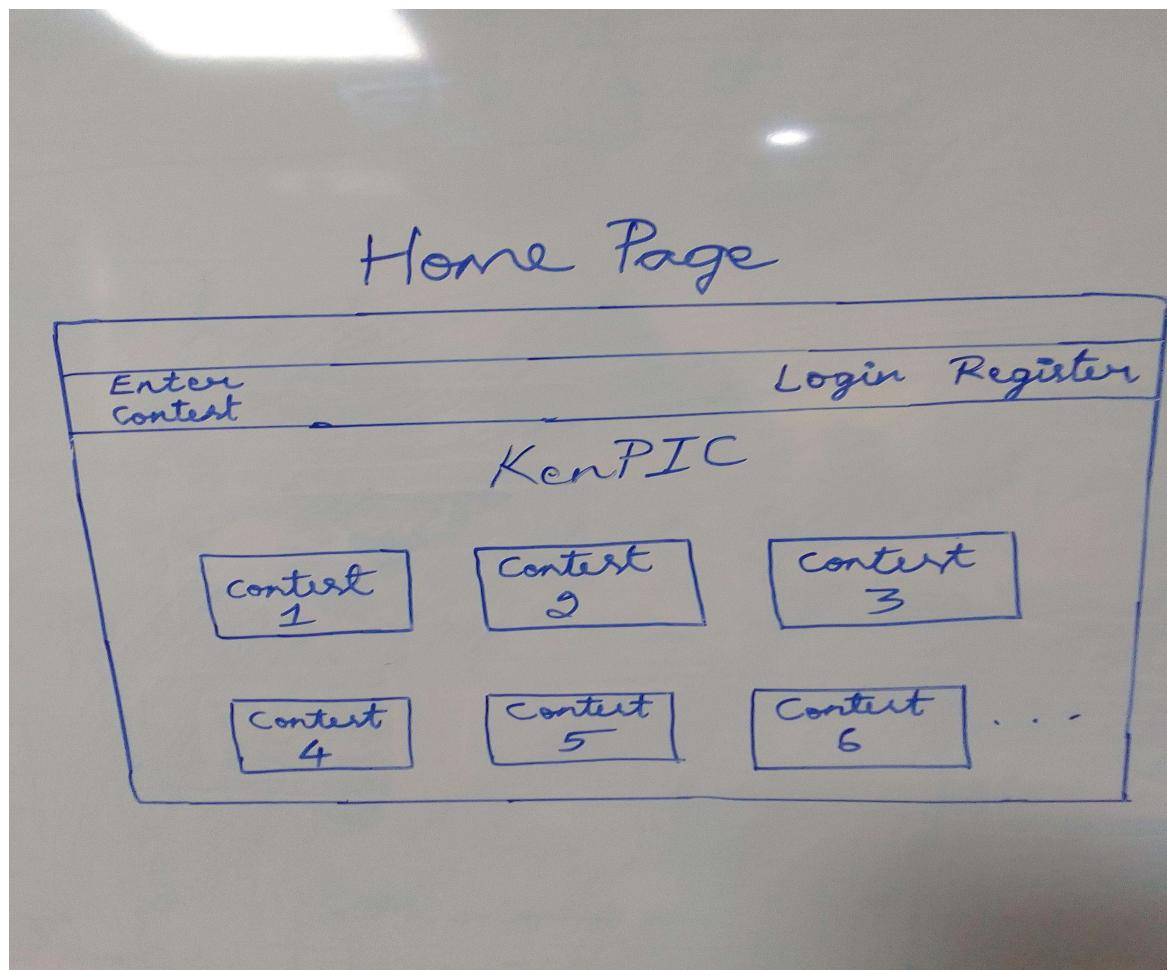


KenPIC

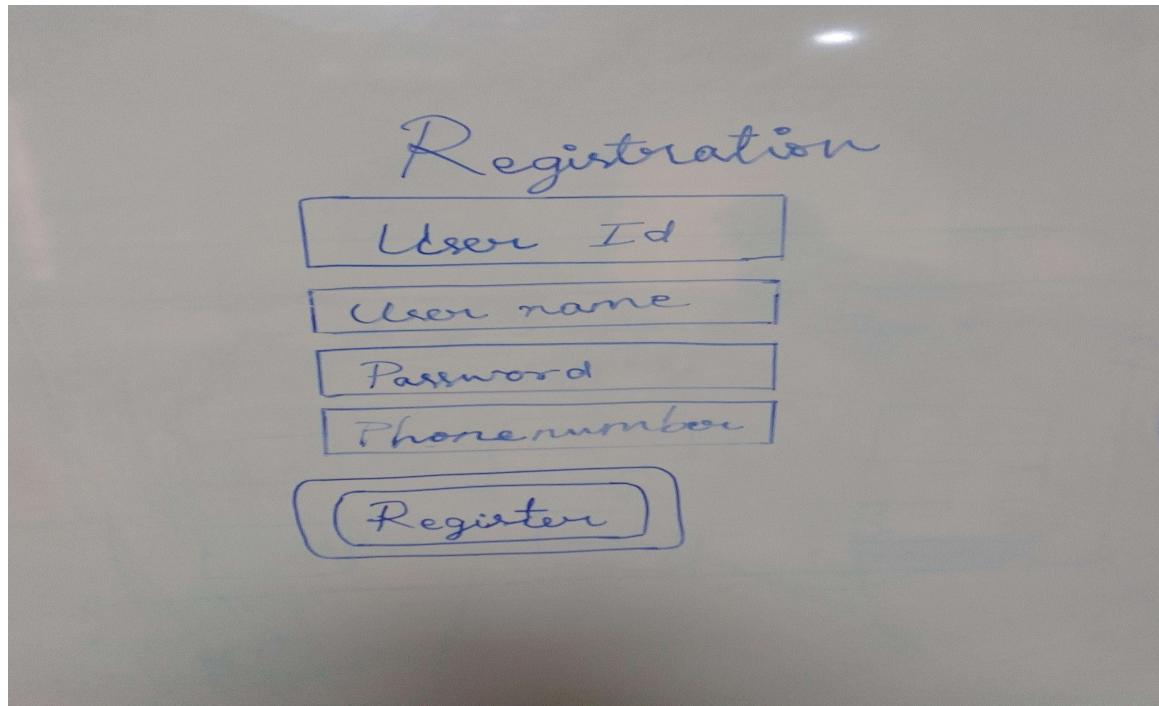
Design Layout

This is the home page of KenPIC portal.

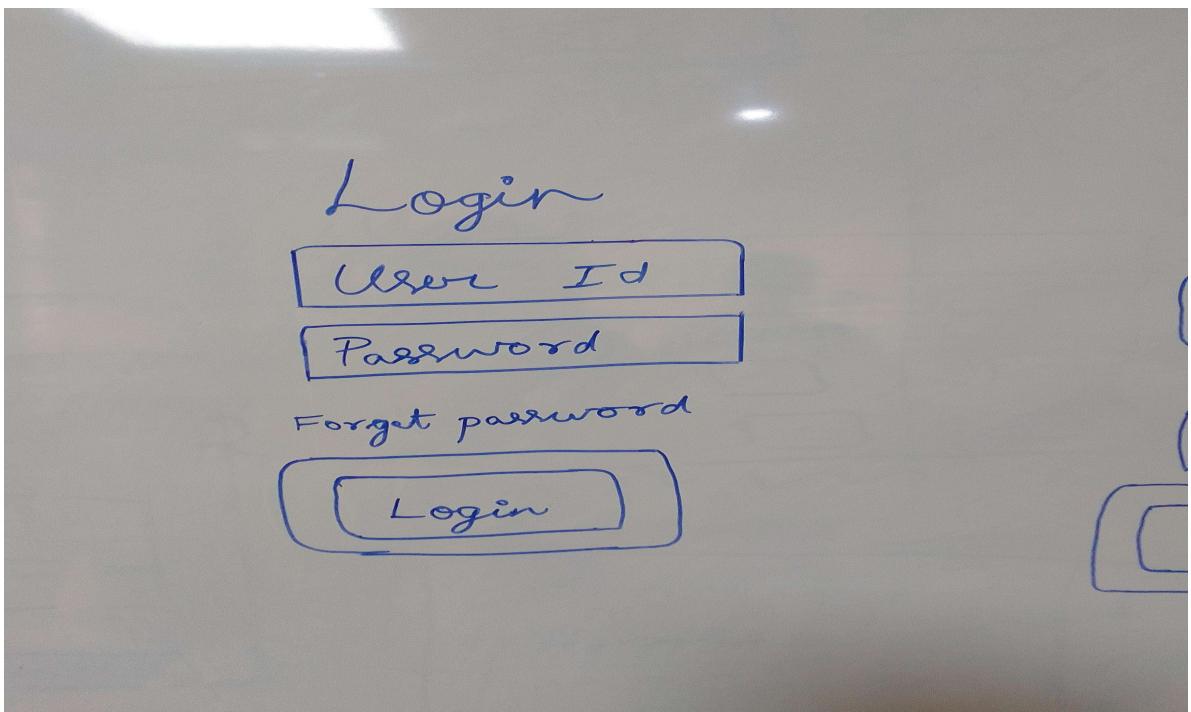


q

This is the page for user to register on the website.



Registered User logs into the portal using this page.



If user clicks forgot password, portal leads to this page.

Forget Password

Phone number

Get OTP

Enter OTP

Verify

Once the OTP is verified, user can reset the password here.

Reset Password

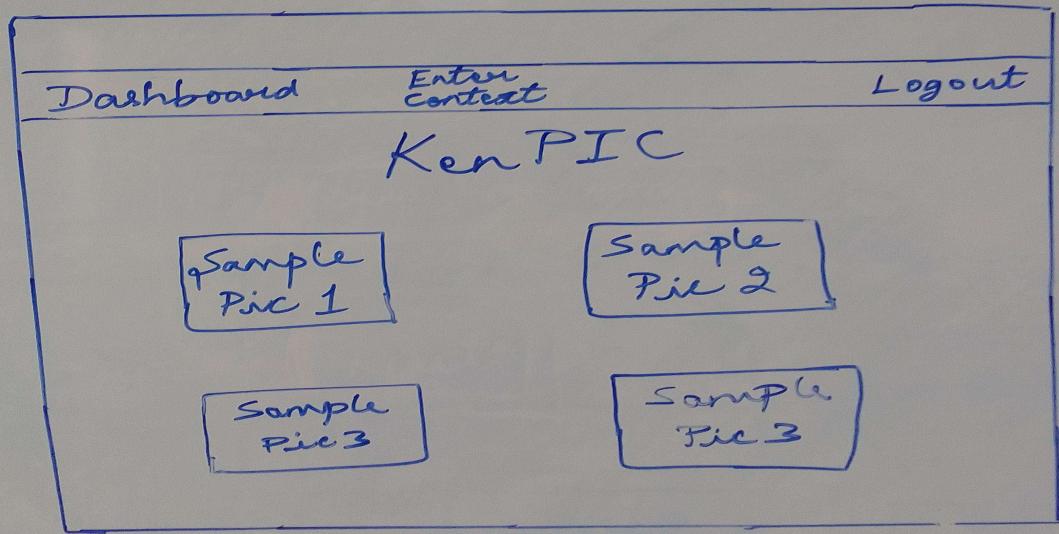
Enter New Password

Confirm Password

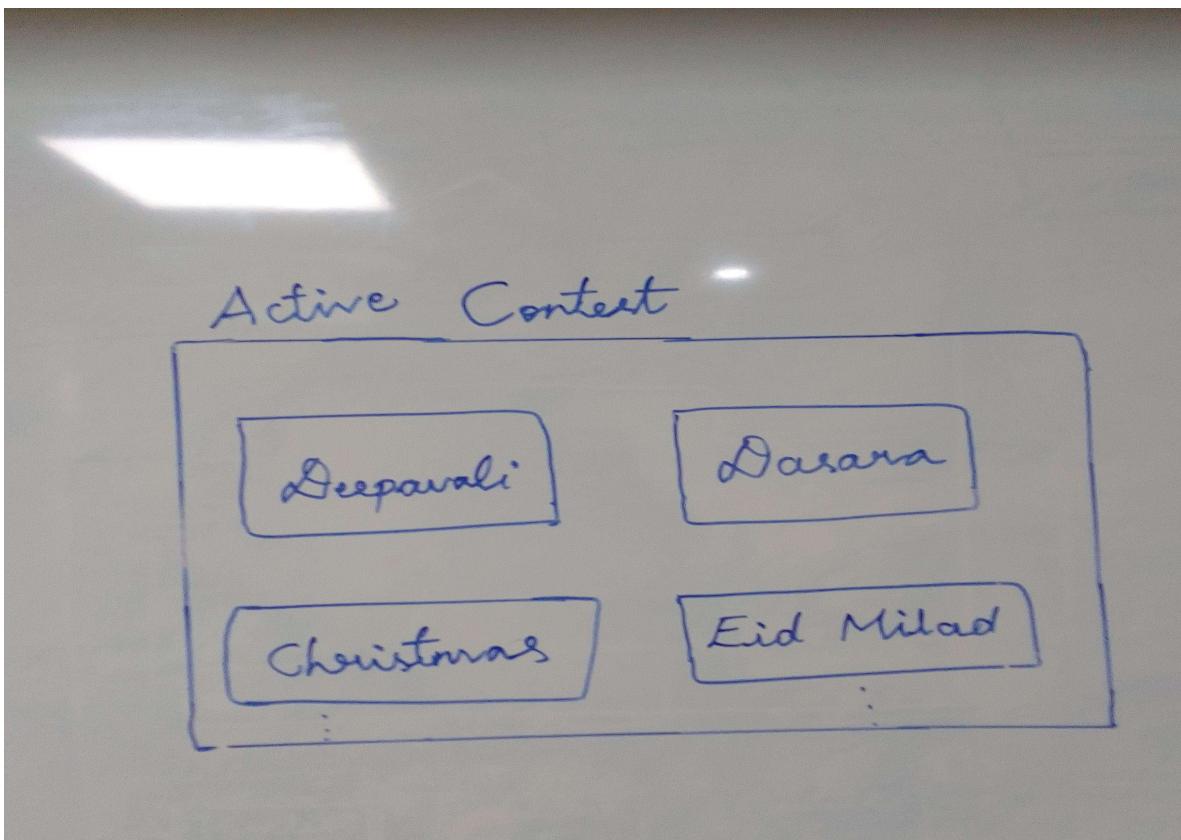
Reset

Once the user logs in to the portal, this is the homepage that will be displayed.

User Home Page



On clicking enroll contest, this page containing all active contest for the time zone will be displayed.



User can view his details, the contest he had and has enrolled and their corresponding results will be displayed here.

User Dashboard

- User Id

- User name

- List of previous contest

user name	image title	Total score
-----------	-------------	-------------

:	:	:
---	---	---

Here Judge can assign score.

Judge

Assign Score

When the admin logs in this page is displayed which can be accessed by admin only.

Admin Page

Admin

- Assign Role
- Create contest
- Get all users
- Delete image
- Search users by Name
- Search users by user id
- Update contest Details
- Report
 - Number of users enrolled in contest
 - Number of photos upload in contest
- Assign Awards for contest
- Fetch top performer of a contest
- Change Judge

API Design

1. User Controller

□ **Register User:** Registers a new User in the system.

- Endpoint: POST /users/register
- Request body:
 - **userid**(string,required): Sasken employee id of the User.
 - **username** (string, required): The username of the new User.
 - **password** (string, required): The password of the new User.
 - **phononenumber**(long, required): Phone number of the User.
- Response:
 - Status code 200 Ok: User is created successfully.
 - Status code 400 Bad Request: Invalid request body.
 - Status code 409 Conflict: The userid and password already exists.

□ **User Login:** Logs in user into the portal

- Endpoint: POST/users/login
- Request body:
 - **userid**(string,required): Sasken employee id of the User.
 - **password** (string, required): The password of the new User.
- Response:
 - Status code 200 Ok: User logged in sucessfully.
 - Status code 400 Bad Request: Invalid request body.

□ **Change user details:** Once the user has logged in, user can update details.

- Endpoint: PUT /users/changeDetails
- Request body:
 - **userid**(String, required): userid of the User.
 - **username**(String, optional): updated name of user.
 - **phonenumbers**(long, optional): New phone number of user.
- Response:
 - Status code 200 Ok: Details changed successfully.
 - Status code 404 Not found: Provided userid was not found in database.

□ **Get user details:** This returns all the user details.

- Endpoint: GET /users/getUserDetails
- Request body:
 - **userid**(String, required): userid of the User.
- Response:
 - Fetches all user details except password.
 - Status code 200 Ok: Details changed successfully.
 - Status code 404 Not found: Provided userid was not found in database.

□ **Get user details By Pattern:** This searches for the user whose name matches the input pattern.

- Endpoint: GET /users/searchUser
- Request body:

- **pattern**(String, required): pattern that depicts username.
- Response:
 - Fetches user details with the provided name except password.
 - Status code 200 Ok: Details returned successfully.
 - Status code 404 Not found: Provided username was not found in database.

2. Image controller

- **Upload Image:** The user enrolled for a contest can upload image.
- Endpoint: POST /image/uploadImage
- Request body:
 - **contestName**(String, required): Name of the contest that the user has enrolled.
 - **imageTitle**(String, required): Name of the image.
 - **image**(String, required): Url of the image to be uploaded.
- Response:
 - Status code 200 Ok: image uploaded successfully.
 - Status code 404 Not found: contest Name was not found in database.

Delete Image: user can delete image uploaded.

- Endpoint: DELETE /image/deleteImage
- Request body:
 - **imageId**(String, required): imageId of the image uploaded.
- Response:
 - Status code 200 Ok: image deleted successfully.

- Status code 404 Not found: imageUrl was not found in database.
- **Fetch Images:** Returns images that were judged by particular judge in a particular contest.
 - Endpoint: GET /image/getImagesByContestTypeandJudgeId
 - Request body:
 - **contestType**(String, required): name of the contest.
 - **JudgeId**(String, required): userId of the judge.
 - Response:
 - Returns list of images based on type of contest and judgeId in the particular contest.
 - Status code 200 Ok: images returned successfully.
 - Status code 404 Not found: invalid request.
- **Fetch Images By Contest:** Returns all the images uploaded for a particular contest type.
 - Endpoint: GET /image/getImagesByContestType
 - Request body:
 - **contestType**(String, required): name of the contest.
 - Response:
 - Returns images uploaded for a particular contest
 - Status code 200 Ok: images returned successfully.

- Status code 404 Not found: invalid request.

3. Contest Controller

□ **Create Contest:** Admin can create a new Contest.

- Endpoint: POST /contest/createContest
- Request body:
 - **contestName**(String, required): Name of the contest.
 - **startDate**(LocalDateTime, required): Date when the contest starts.
 - **endDate**(Datetime, required): date when the contest ends.
- Response:
 - Status code 200 Ok: Contest created successfully.
 - Status code 400 Bad Request: Invalid request body.

□ **Update Contest details:** Admin can update contest details.

- Endpoint: PUT /contest/updateContest
- Request body:
 - **contestName**(String, required): Name of the contest.
 - **startDate**(LocalDateTime, optional): Date when the contest starts.
 - **endDate**(Datetime, optional): date when the contest ends.
- Response:
 - Status code 200 Ok: Contest updated successfully.
 - Status code 400 Bad Request: Invalid request body.

- **Get Active contest:** Retrieves a list of all contests that are active which will be visible to admin.
 - Endpoint: GET/contest/getContestByDate
 - Request body:
 - **startDate**(LocalDateTime, required): Date when the contest starts.
 - Response:
 - Returns the list of all contests that start at the provided date.
 - Status code 200 Ok: All active Contests retrieved successfully.
 - Status code 400 Bad Request: Invalid request body.

- **Search contests:** Returns the list of all the contests using contest Name.
 - Endpoint: GET/contest/searchContest
 - Request body:
 - **contestName**(String, required): The contest name which has to be searched.
 - Response:
 - Returns the contest from name provided.
 - Status code 200 Ok: Contest retrieved successfully.
 - Status code 400 Bad Request: Invalid request body.

4. Score Controller

- **Assign Score:** Judge assigns score to images in particular contest.
 - Endpoint: POST/score/assignScore
 - Request body:

- **judgeId**(String, required): userid of the judge.
 - **ImageId**(String, required): imageId of the image.
- Response:
 - Status code 200 Ok: score assigned successfully.
 - Status code 400 Bad Request: Invalid request body.

□ **Fetch score of user for a contest:** It returns the total score of the user in particular contest.

- Endpoint: GET/score/getscoreByUseridandContestName
- Request body:
 - **userid**(String, required): userid of the user.
 - **ContestName**(String, required): name of the contest.
- Response:
 - Status code 200 Ok: score assigned successfully.
 - Status code 400 Bad Request: Invalid request body.

5. Admin Controller

- **Number of images in contest:** Returns number of images uploaded for a particular contest.
- Endpoint: GET/admin/getNumberofImagesByContestName
 - Request body:
 - **contestName**(String, required): name of the contest.
 - Response:

- Returns number of photos uploaded for a contest.
- Status code 200 Ok: Response returned successfully.
- Status code 400 Bad Request: Invalid request body.

□ **Number of users enrolled for a contest:** Returns number of user enrolled for a particular contest.

- Endpoint: GET/admin/getNumberOfUsersByContestName
- Request body:
 - **contestName**(String, required): name of the contest.
- Response:
 - Returns number of users enrolled for a contest.
 - Status code 200 Ok: response returned successfully.
 - Status code 400 Bad Request: Invalid request body.

◊ Awards

□ **Get Top Performers:** Returns imageId of top 3 performers in particular contest.

- Endpoint: GET/admin/getImageIdByScoreinDesc
- Request body:
 - **contestName**(String, required): name of the contest.
- Response:
 - Returns top 3 high sored images contest.

- Status code 200 Ok: response returned successfully.
- Status code 400 Bad Request: Invalid request body.

□ **Update award won:** The imageId which won the award for particular contest , in that corresponding score table award will be updated.

- Endpoint: PUT/admin/updateAward
- Request body:
 - **contestName**(String, required): name of the contest.
 - **ImageId**(String, required): the image which won the award.
- Response:
 - Status code 200 Ok: awards updated successfully.
 - Status code 400 Bad Request: Invalid request body.

Assign judge:

Change role user:

ER Diagram

