

Fig 1: Home Page

Home Page :-

The home page function in a **platform for Assessing Job Skills** web application renders the home.html template when a request is made. It takes the request object as a parameter and returns the rendered template. This function serves to display the home page of the web application. Non-authenticated users would only see "Login" and "Register" links. This approach simplifies the menu by treating all logged-in users the same, with differentiating between regular users and staff members. It ensures that all authenticated users have access to the same features, streamlining the user interface.

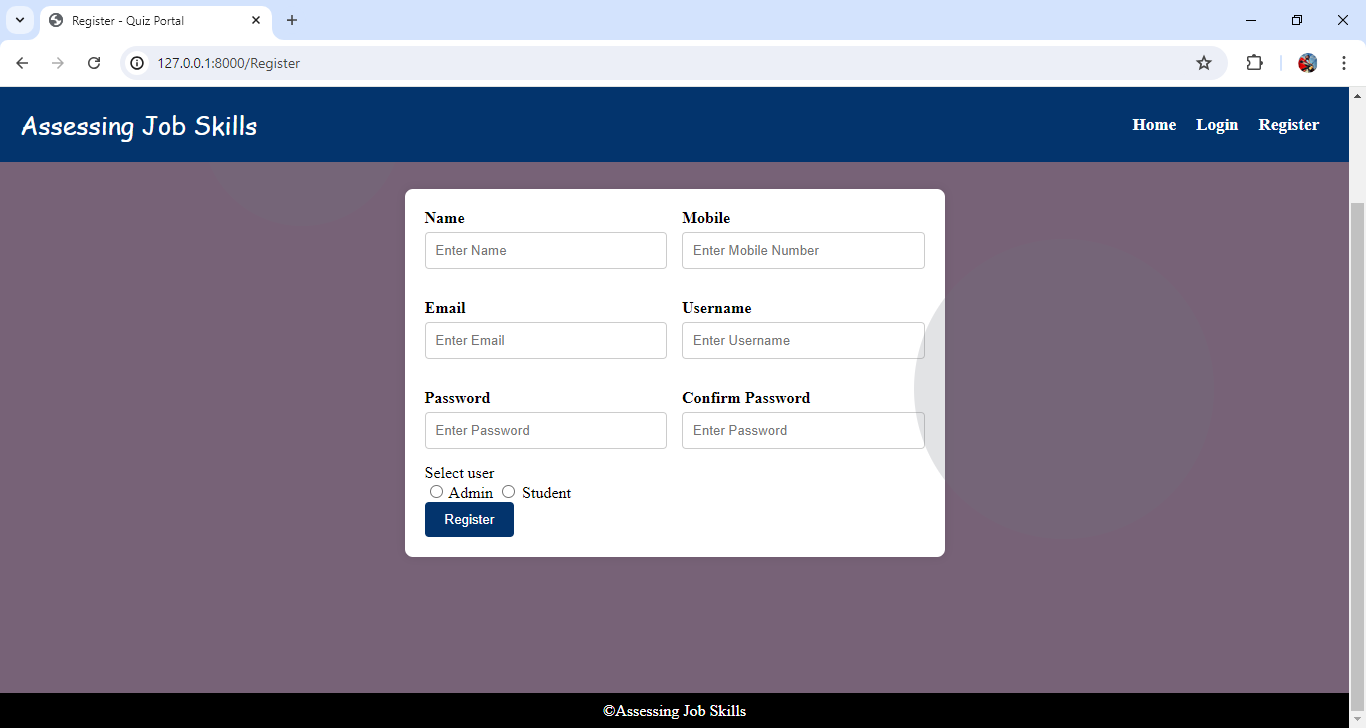


Fig 2: Registration Page

REGISTER :-

The register function handles user registration in a **platform for Assessing Job Skills** web application. When a POST request is made, it retrieves user details from the form, including name, email, username, password, confirmation password, and user type (admin or regular). It checks if the passwords match and whether the username already exists. If the username is unique and passwords match, a new user is created with the provided details, including setting the user as staff if selected. On success, it redirects to the login page with a success message. If there are errors, appropriate error messages are displayed, and the user is redirected back to the registration page. For GET requests, it renders the registration form.

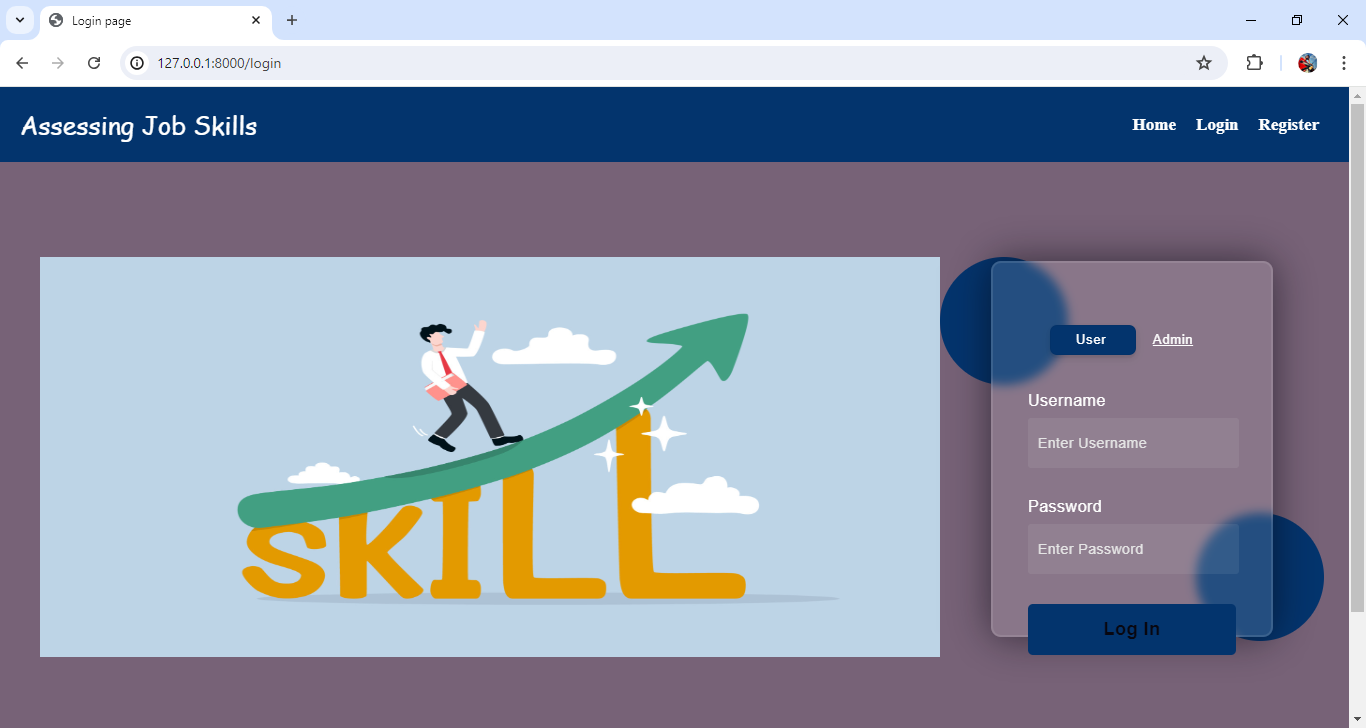


Fig 3: Login For Admin and User

LOGIN :-

The login function handles user authentication in a **platform for Assessing Job Skills** web application. It processes POST requests by retrieving the username and password, authenticates the user, and logs them in if the credentials are correct. On successful login, it redirects to the home page and shows a success message. If authentication fails, it redirects back to the login page with an error message. For GET requests, it renders the login page.



Fig 4: Admin Home Page

Admin Home Page :-

The navigation menu would display the same options for all authenticated users. Logged-in users would see links to "Skills," "create Skill," "Add Questions," and "Logout," regardless of their role or privileges. Non-authenticated users would only see "Login" and "Register" links. This approach simplifies the menu by treating all logged-in users the same, with differentiating between regular users and staff members. It ensures that all authenticated users have access to the same features, streamlining the user interface.

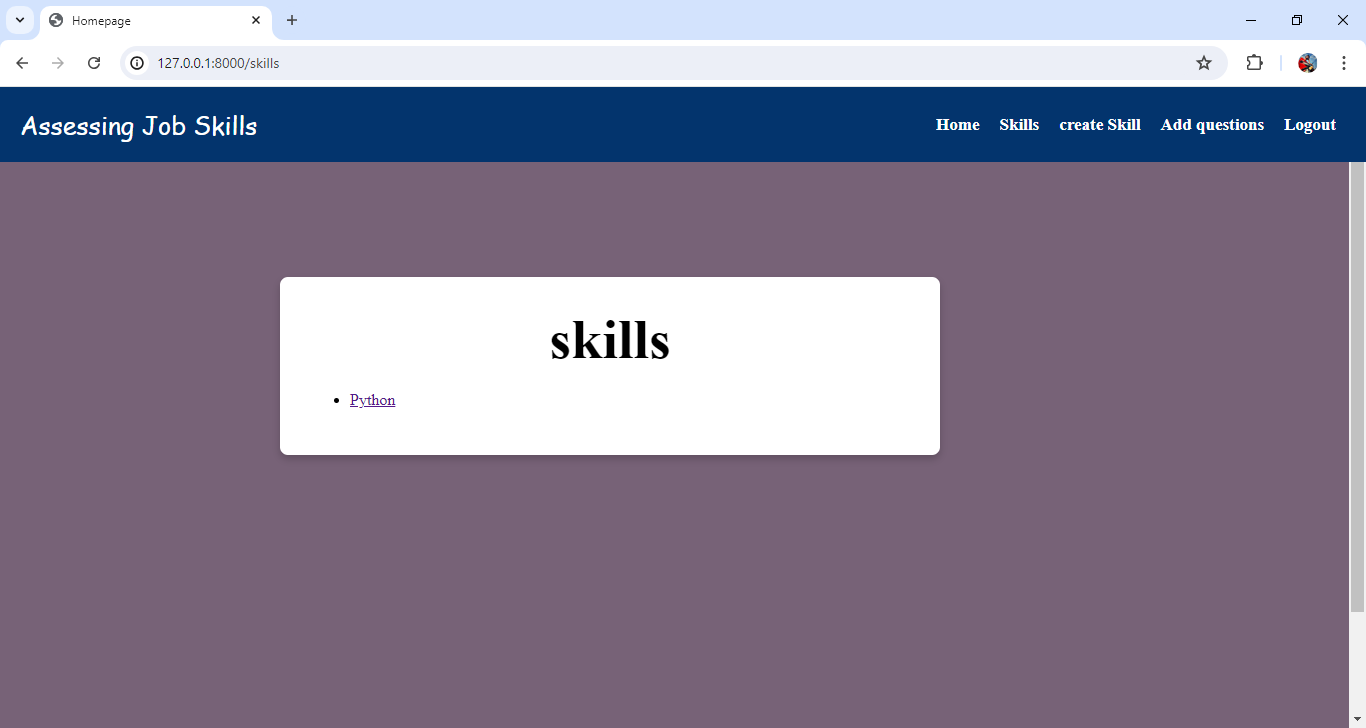


Fig 5: Existing Skills Page

The Skills page retrieves all skill objects from the database using Function and stores them in the skills variable. It then renders the skill template, passing the retrieved skills to the template for display. This allows administrators to view and manage all skills on the skills Page.

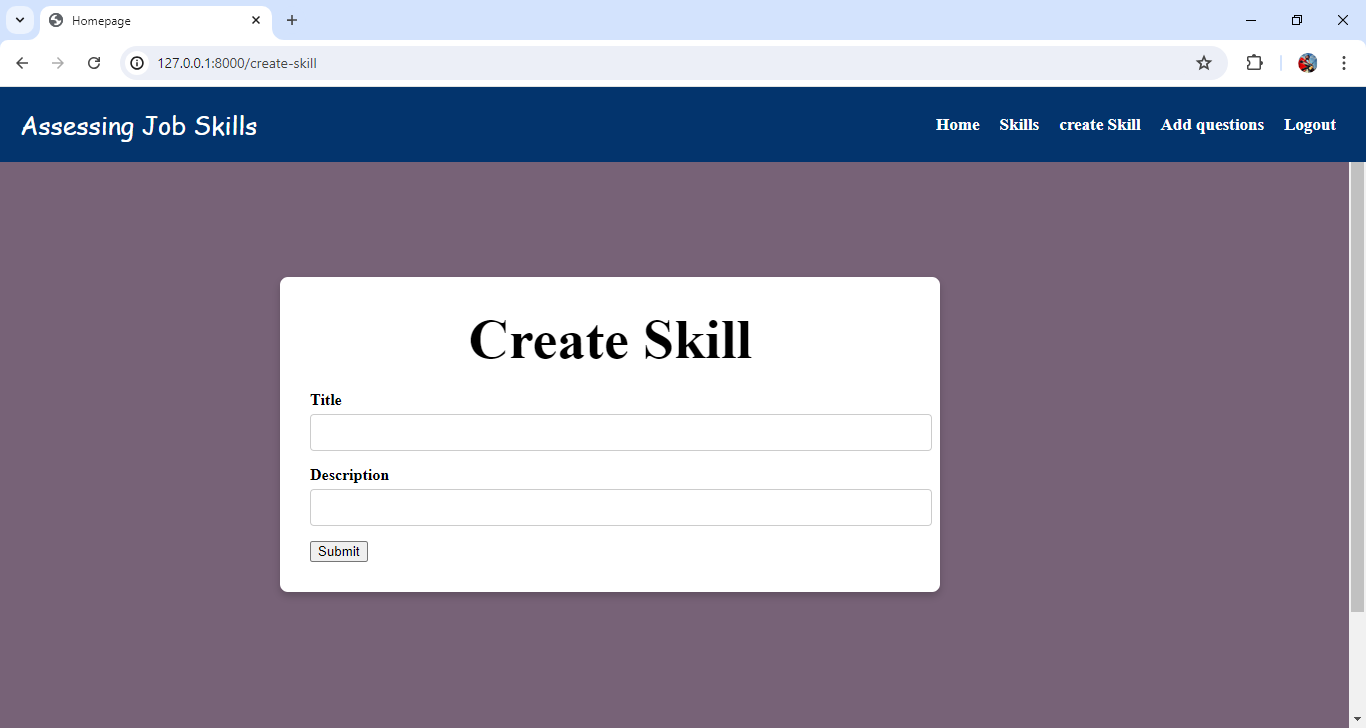


Fig 6: create Skill

Create Skills:-

The create Skill page function handles the creation of new skills. When a POST request is received, it retrieves the skill title and description from the form data and creates a new skill object with these details. After saving the new skill to the database, it renders the Create Skill template. This setup allows users to input skill details through a form and add new skills to the system.

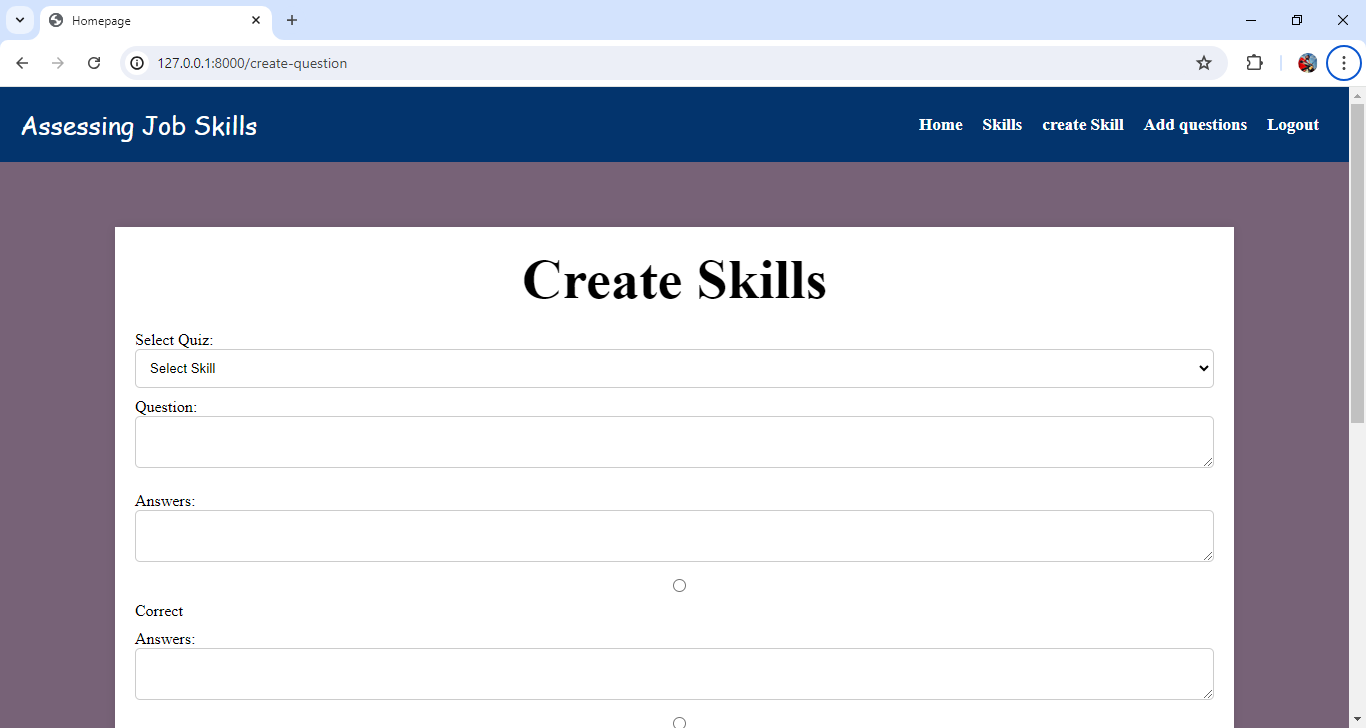


Fig 7: creating Questions for Skill

Add Questions :-

The Add Question page function manages the creation of new questions for skills. It first retrieves all available skills to populate a dropdown or selection menu. If a POST request is received, it extracts the question text, selected skill ID, and answers from the form data. It then creates a new Question object associated with the selected quiz and saves it. Following this, it fetches all questions for the selected skill and processes the provided answers, marking one as correct based on the index supplied. Each answer is saved in the database with a flag indicating whether it is correct. Finally, it renders the Add Question Page template, passing the list of skills and the newly created questions for display



Fig 8: User Home Page

User Home Page :-

The navigation menu would display the same options for all authenticated users . Logged-in users would see links to "Home," "skill," and "Logout," regardless of their role or privileges. Non-authenticated users would only see "Login" and "Register" links. This approach simplifies the menu by treating all logged-in users the same, with differentiating between regular users and staff members. It ensures that all authenticated users have access to the same features, streamlining the user interface.

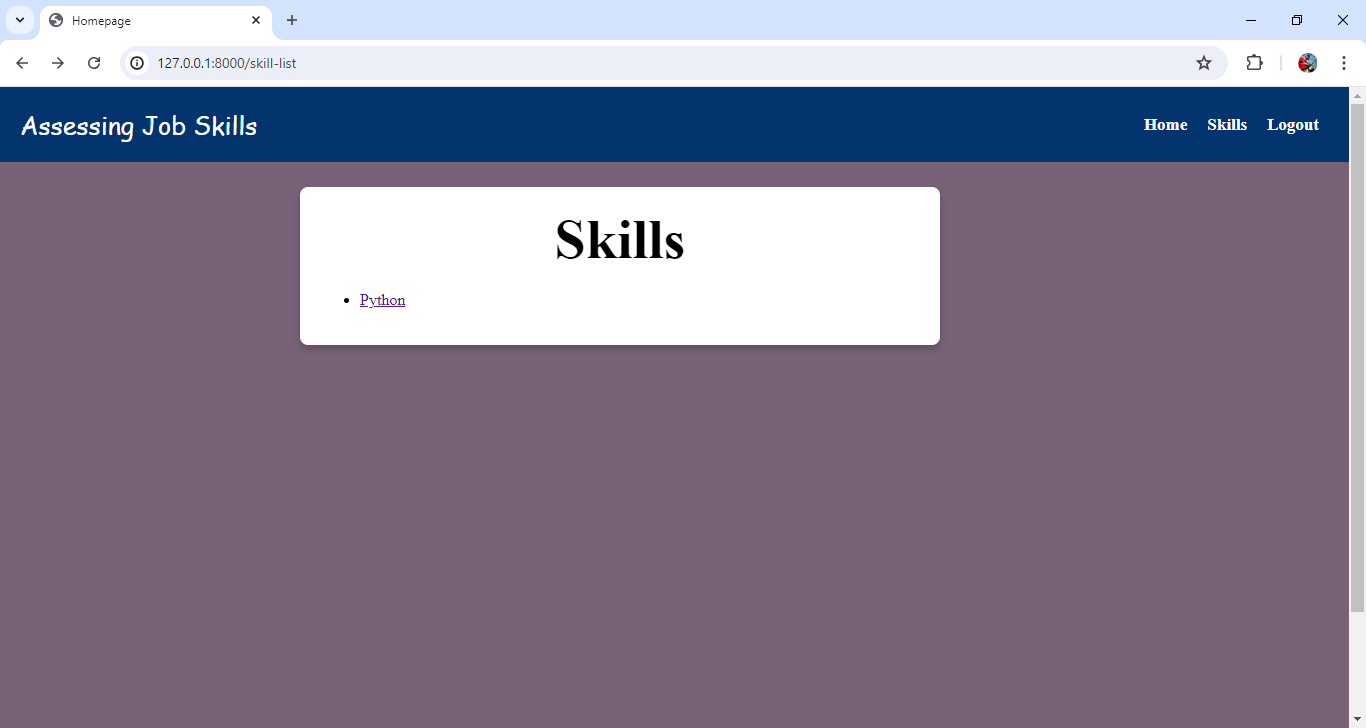


Fig 9: User skill Dashboard click on skill title to start the task

Skills :

The Skills page function retrieves all skill objects from the database and passes them to the skill page template. In the template, each skill is displayed with a link (e.g., <a href="">) that redirects users to a page where they can start the test for that specific skill. This setup allows users to view a list of available quizzes and easily navigate to begin taking any selected skill.

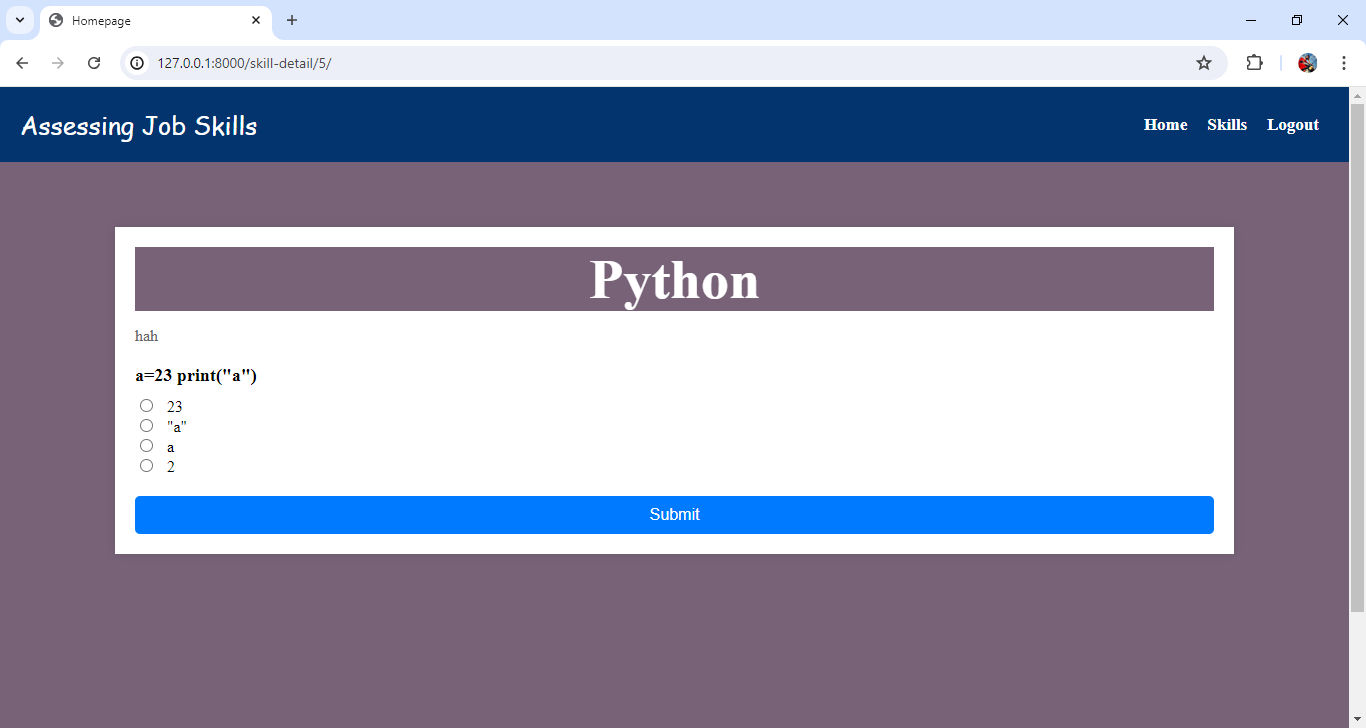


Fig 10: Test Questions

Skill page (Skill Assessment start):-

The Skills page function handles the display and submission of Skills questions. It retrieves a specific Skill object based on the primary key (pk) and fetches associated questions. If the request method is POST, it calculates the user's score by iterating through each question, checking if the selected answer is correct, and updating the score accordingly. After calculating the score, it creates a Score object to record the user's performance and saves it. The function then renders the score Page template.

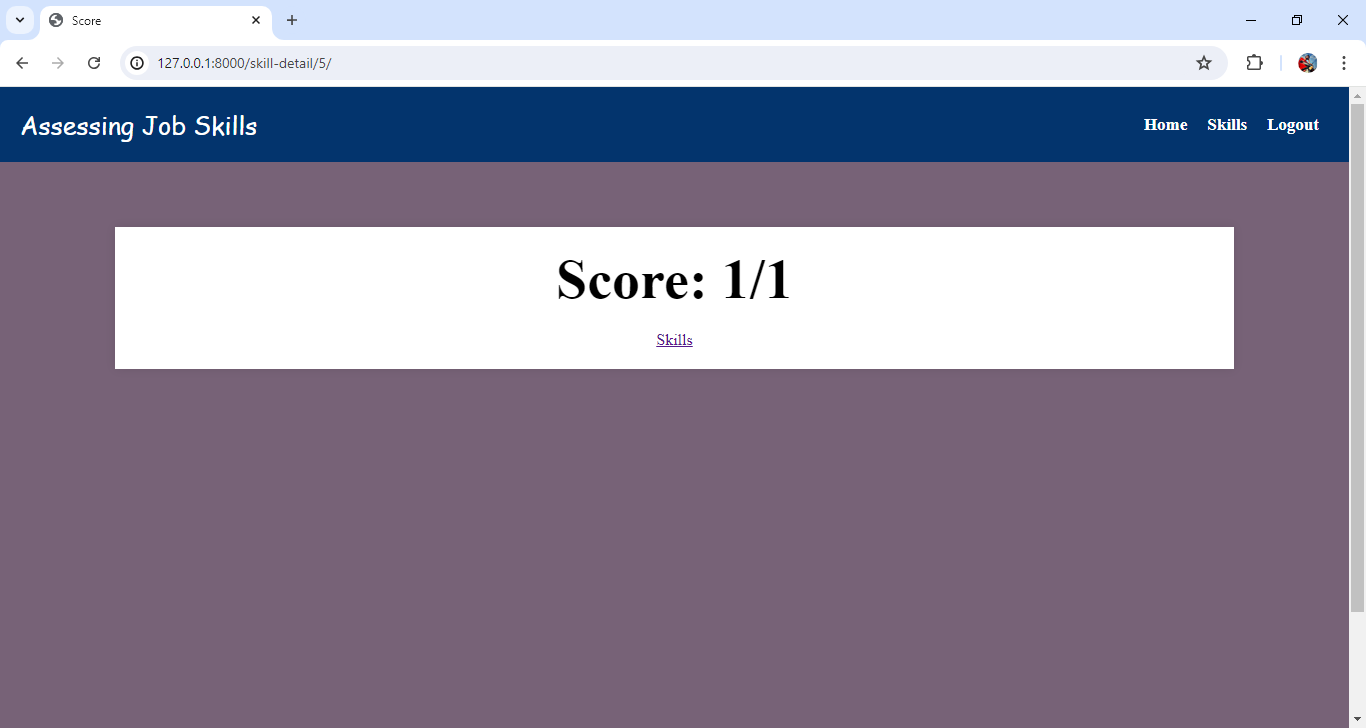


Fig 11: Score Result

Score Page:-

The Score page function renders the score to display the user's score and total number of questions from the skill template. For GET requests, it renders the Skill template, showing the quiz and its questions.