Pages (KEKO24-56) [KEKO24-97] CreateJobPage Created: 11/May/25 Updated: 23/May/25 Resolved: 11/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Pages

Type:	Story	Priority:	Medium			
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified					
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Sprint:	KEKO24 Sprint 3
Rank:	0 i000kf:

Description

User Story: Post a New Job on the Create Job Page

As a registered client,

I want to access a Create Job page with a form,

So that I can post a new job and provide all necessary details for freelancers to view and apply.

Acceptance Criteria:

AC1 – Access to Create Job Page

- Given the client is logged in and on their profile or dashboard,
- When they click the "Post a Job" button,
- Then they should be redirected to the **Create Job** page.

AC2 – Display Job Creation Form

- The page should display a form with the following fields:
 - Job title (text input)
 - o Job description (textarea)
 - o Category (dropdown or select)
 - Required skills (tags or multi-select)

- o Location (optional country or remote)
- o Budget
- o Deadline (date picker optional)

AC3 – Form Validation

- All required fields should be validated before submission.
- If a required field is missing or invalid, a helpful error message should be shown.

AC4 – Submit Job Post

- When the client fills in the form and clicks the "Post Job" button,
- Then the job should be saved to the database,

Pages (KEKO24-56)

Parent:

[KEKO24-96] JobsListPage Created: 11/May/25 Updated: 23/May/25 Resolved: 11/May/25

Status: Done
Project: Keko 2024
Components: None
Affects versions: None
Fix versions: None

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 3
Rank:	0 i000k7:

Description

User Story: View All Job Listings on JobsListPage

As a user interested in exploring more job opportunities,

I want to navigate to a dedicated Job Listings page,

Pages

So that I can browse through all available job postings in one place.

Acceptance Criteria:

AC1 – Navigate from Job Section

- Given the user is on the homepage (or relevant section with limited job listings),
- When they click the "More Jobs" button,
- Then they should be redirected to the **JobsListPage**.

AC2 – Display All Job Listings

- Given the user is on the **JobsListPage**,
- Then the page should fetch and display all available job postings.

AC3 – Job Post Details

- Each job card/list item should include:
 - Job title
 - Location (city/country)
 - o Budget
 - o (Optional: date posted, employer name)

AC4 – Consistent Layout and Pagination (Optional)

• The listings should be presented in a clean, scrollable or paginated format to ensure good usability.

Pages (KEKO24-56) [KEKO24-95] ServicesPage Created: 11/May/25 Updated: 11/May/25 Resolved: 11/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Pages

Type:	Story	Priority:	Medium	
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby	
Resolution:	Done	Votes:	0	
Labels:	None			
Remaining Estimate:	Not Specified			
Time Spent:	Not Specified			
Original estimate:	Not Specified			

Sprint:	KEKO24 Sprint 3
Rank:	0 i000jz:

Pages (KEKO24-56) [KEKO24-94] AboutPage Created: 11/May/25 Updated: 11/May/25 Resolved: 11/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Pages

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 3
Rank:	0 i000jr:
Participants:	Maryam Agaiby

Navigation and routing (KEKO24-92) [KEKO24-93] Implement navigation for header items Created: 05/May/25 Updated: 05/May/25			
Status:	To Do		
Project:	Keko 2024		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Parent:	Navigation and routing		

Type:	Subtask	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Attachments:	image-20250505-081025.png	
Sprint:	KEKO24 Sprint 3	
Rank:	0 i000jj:	

Description

THere are lot of links to other pages in header and the footer. Implement navigation for header.

If there is a navigation link that has no frontend implementation then add dummy file for that route/compound



Wrapper Components (KEKO24-55)

[KEKO24-92] Navigation and routing Created: 05/May/25 Updated: 11/May/25

Status: In Progress
Project: Keko 2024
Components: None

Affects versions: None

Fix versions: None

Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Maryam Agaiby
Resolution:	Unresolved	Votes:	0
Labels:	None		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key Summary		Type	Status	Assignee
	KEKO24-93	Implement navigation for header items	Subtask	To Do	
Sprint:	KEKO24 Sprint 3				
Rank:	0 i000jb:				
Participants:	Maryam Agaiby, Uzair Irshad				

Description

How to organize pages and routes in React app with react router 7?

How user can navigate to a new page to fill a form and then automatically be returned to previous page?

If there is a list view and user can add new item to the list how dialog/page to add new item is handled and how the user is redirected back to the list and list is updated with the newest addition?

Wrapper Components (KEKO24-55) [KEKO24-91] CreateJob & JobsList & Jobs'Mangement Created: 04/May/25 Updated: 23/May/25 Status: In Progress Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Attachments:	□ jobForm1.png □ jobForm2.png
Sprint:	KEKO24 Sprint 3
Rank:	0 i000hz:
Participants:	Maryam Agaiby

Description

User Story 1: Create Job Posting via UI

As an employer,

I want to fill out a form to create a new job posting, so that I can publish available positions through the system.

Acceptance Criteria:

• AC1: Display job creation form

- o Given the employer is logged in and on the "Create Job" page,
- When the page loads,
- o *Then* the form should include fields for title, description, budget, location, start date, end date, and job status.

• AC2: Validate required fields and input types

- o Given the employer is filling out the job creation form,
- o When required fields are left empty or data types are invalid (e.g., budget is not a number),
- o *Then* the form should display validation messages and disable submission.

• AC3: Submit new job posting

o Given the form is valid,

- o When the employer clicks the "Submit" button,
- o *Then* the frontend should call the POST /jobs API and display a success message on response.

User Story 2: View and Manage Job Postings

As an employer,

I want to see a list of jobs I've created, so that I can view or update postings as needed.

Acceptance Criteria:

• AC1: Display list of job postings

- o Given the employer is logged in and navigates to the "Dashboard" page,
- When the page loads,
- o *Then* the system should fetch and display all job postings created by the employer using the GET /jobs API.

• AC2: Navigate to edit a job

- o Given the employer is on the "Dashboard" page,
- When they click an "Edit" button for a job,
- o *Then* they should be taken to a form where they can update job details.

• AC3: Update job details

- o Given the employer is editing a job,
- o When they modify fields and submit the form,
- o *Then* the frontend should send a PUT /jobs/:id request and display a success confirmation upon completion.

User Story 3: Archive (Soft Delete) Job Postings

As an employer,

I want to archive job postings, so that they are no longer visible to applicants but still stored in the system.

Acceptance Criteria:

• AC1: Archive job from job list

- o Given the employer is on the "My Jobs" page,
- o When they click the "Archive" button on a job listing,
- o *Then* the frontend should call DELETE /jobs/:id and update the UI to reflect the inactive status.

• AC2: Archived jobs are visually distinct

- o Given a job is archived,
- o When it appears in the job list (if shown),
- o Then it should be visually marked as "Archived" or "Inactive."

Backend Services (KEKO24-83)

[KEKO24-90] Resume upload Created: 28/Apr/25 Updated: 05/May/25

Status:	In Progress
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Parent: Backend Services

Type:	Story	Priority:	Medium
Reporter:	Rizwan Anwar	Assignee:	Rizwan Anwar
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 3
Rank:	0 i000hr:

Description

User Story:

As a backend developer,

I need to implement a system to allow users to upload their resume directly to the server, so that the system can store a single resume file for each user's profile or job application.

Acceptance Criteria

AC1: Enable Direct Resume Upload to Server

Given an authenticated user uploads a resume file through a form or interface,

When the upload is initiated,

Then the resume file is saved directly to the server storage (e.g., a designated folder) with the correct user association.

AC2: Validate Input Data

Given the user uploads a resume,

When the file is missing or in an invalid format (e.g., not PDF or DOCX) or exceeds size limits, Then the system shows an error message and prevents the upload.

AC3: Ensure Authentication

Given a user is not logged in,

When they attempt to upload a resume,

Then the system rejects the upload and prompts them to log in.

AC4: Allow Only One Resume Upload per User

Given a user has already uploaded a resume,

When they try to upload another resume,

Then the system either rejects the new upload with an error message or overwrites the existing resume based on business rules.

Backend Services (KEKO24-83) [KEKO24-89] User Management Created: 13/Apr/25 Updated: 12/May/25 Resolved: 12/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None

Type:	Story	Priority:	Medium		
Reporter:	Muhammad Rashid	Assignee:	Muhammad Rashid		
Resolution:	Done	Votes:	0		
Labels:	None	None			
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Sprint:	KEKO24 Sprint 3	
Rank:	0 i000ek:	
Participants:	Anastasiia, Muhammad Rashid	

Description

Parent:

User management and user authentication

Backend Services

Backend Services (KEKO24-83) [KEKO24-88] Jobs Created: 07/Apr/25 Updated: 23/May/25 Resolved: 11/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None

Type:	Story	Priority:	Medium
Reporter:	<u>Uzair Irshad</u>	Assignee:	Maryam Agaiby
Resolution:	Done	Votes:	0
Labels:	None		·
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original	Not Specified		

Sprint:	KEKO24 Sprint 3	
Rank:	0 i000ei:	
Participants:	Maryam Agaiby, Uzair Irshad	

Description

estimate:

Fix versions:

Parent:

None

Backend Services

User Story: Create and Manage Job Records in the System

As an employer,

I want to create and manage job postings with detailed information, so that I can effectively publish available positions and manage the hiring process.

Acceptance Criteria

AC1: Create a new job record in the system

- **Given** the employer is authenticated and authorized,
- When they provide valid job data including title, description, budget, location, dates, and status,
- **Then** a new job entry should be created in the database with associated metadata (createdBy, createdAt).

AC2: Enforce required fields and data types

• **Given** a request to create or update a job,

- When required fields (title, jobStatus, startDate, endDate, createdBy) are missing or data types are invalid,
- Then the API should respond with a validation error and not persist the record.

AC3: Job records are retrievable and updatable

- **Given** a valid job ID,
- When the job is queried or updated by an authorized user,
- Then the system should return or update the job with accurate data including all associated fields.

AC4: Soft delete or deactivate job listings

- **Given** an employer wants to remove a job from being listed,
- When they perform a delete or archive action,
- **Then** the job should be marked as inactive without losing its historical data in the database.

Technical Notes (for internal dev reference)

- Database Table: jobs
- Technology Stack: NestJS, TypeORM, PostgreSQL
- Fields:
 - o id: Primary Key (UUID)
 - o createdBy: Foreign Key (users.id)
 - o createdAt: Timestamp
 - o title: String
 - o jobStatus: Enum (e.g., OPEN, IN PROGRESS, CLOSED)
 - o description: Text
 - o startDate: Date
 - o endDate: Date
 - o budget: string
 - o location: String or JSONB (for structured location data)

• Relationships:

- o createdBy should be a foreign key referencing the users table.
- o Include decorators like @ManyToOne, @CreateDateColumn, etc.

• Validation:

- Use class-validator decorators in DTOs.
- o Ensure startDate ≤ endDate.

• Endpoints (Example):

- o POST /jobs
- o GET /jobs/:id
- o PUT /jobs/:id
- o DELETE /jobs/:id

Backend Services (KEKO24-83) [KEKO24-87] Audit Logging Created: 07/Apr/25 Updated: 07/May/25 Resolved: 03/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None

Type:	Story	Priority:	Medium		
Reporter:	Kseniia Kochetova	Assignee:	Kseniia Kochetova		
Resolution:	Done	Votes:	0		
Labels:	None	None			
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Sprint:	KEKO24 Sprint 3
Rank:	0 i000ak:i
Participants:	Arkar Myint Myat, Kseniia Kochetova

Description

Parent:

User Story: Record Inbound Requests and Actions via Interceptors for Audit Logging

As an administrator,

I want the backend system to automatically record all REST API requests and key user actions using interceptors,

so that I can monitor system usage and ensure security compliance.

Acceptance Criteria

AC1: All REST API requests are intercepted and logged

Backend Services

- **Given** any HTTP request is made to the backend API,
- When the request passes through the NestJS interceptor,
- **Then** the system logs the timestamp, HTTP method, endpoint path, IP address, user agent, and request status code.

AC2: Authenticated user context is included in logs

- **Given** a user is authenticated via Passport,
- When their request is intercepted,
- Then the log entry includes their user ID, email, and role.

AC3: Interceptor captures success and error responses

- Given a request is processed by the controller,
- When the response is either successful or results in an error,
- Then the interceptor logs the outcome, including status code and any error message (if applicable).

AC4: Logs are persisted securely in PostgreSQL

- **Given** a log entry is created,
- When the interceptor processes the request,
- Then the log is saved to a dedicated audit_logs table using TypeORM in an append-only structure.

AC5: Only administrators can retrieve audit logs

- Given an authenticated user makes a GET request to /admin/audit-logs,
- When the user has an admin role,
- Then the API returns the logs with pagination, date filtering, and sorting.

AC6: Sensitive request data is masked or excluded

- **Given** the request contains sensitive fields (e.g., passwords, payment details),
- When the log is created,
- Then these fields are masked or omitted from the stored log data.

AC7: Logs are accessible via a secured admin-only REST endpoint

- **Given** the system is running in production,
- When an admin accesses /admin/audit-logs,
- **Then** the request is authenticated and authorized using Passport, and protected via role guards.

Backend Services (KEKO24-83)

[KEKO24-86] Reviews module Created: 07/Apr/25 Updated: 06/May/25 Resolved: 03/May/25

[IXLIXO24-00]	Created: 07/Apr/23 Opdated: 06/May/23 Resolved: 05/May/23
Status:	Done
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None
Parent:	Backend Services

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Abdul Sami
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 3
Rank:	0 i000al:

Description

Title: Create Backend API for Submitting User Reviews

User Story:

As a backend developer,

I need to implement an API to allow users to submit reviews with multiple rating metrics, so that the system can store feedback for a user after a completed interaction.

Acceptance Criteria

AC1: Create Review API endpoint

- **Given** an authenticated user sends a POST request to /api/reviews,
- When the request includes valid receiverId, qualityRating, deadlineRating, communicationRating, and recommendationRating,
- Then the review is saved in the database with the correct createdBy and timestamp.

AC2: Validate input data

• **Given** the user submits a review request,

- When any required field is missing or contains invalid data (e.g., rating out of range),
- **Then** the system returns a 400 Bad Request with a relevant error message.

AC3: Ensure authentication and authorization

- **Given** a user is not logged in,
- When they attempt to submit a review,
- **Then** the system rejects the request with a 401 Unauthorized status.

AC4: Prevent duplicate reviews for the same transaction (if applicable)

- **Given** a user has already submitted a review for a transaction or user,
- When they try to submit another review for the same context,
- **Then** the system returns a 409 Conflict or updates the existing review based on business logic.

AC5: Store metadata correctly

- Given a review is successfully submitted,
- When it is stored,
- Then the system also saves createdAt, and any optional reviewText if provided.

As an agent I want to post a gig / job in the system so the freelancers can offer themselves for that job, Repeated? (KEKO24-84)

[KEKO24-85] Create NestJS resources (controller, service, entity, module,

dto) Created: 02/Apr/25 Updated: 02/Apr/25

Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None
Domonte	As an agent I want to most a sig / ish in the system so the freelengers con

As an agent I want to post a gig / job in the system so the freelancers can offer themselves for that job, Repeated?

Type:	Subtask	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i000hj:

Description

Create required resources and develop necessary endpoints and so on...

Backend Services (KEKO24-83)

[KEKO24-84] As an agent I want to post a gig / job in the system so the freelancers can offer themselves for that job, Repeated? Created: 02/Apr/25 Updated:

11/May/25 Resolved: 11/May/25

Status: Done

Project: Keko 2024

Components: None

Affects versions: None

Fix versions: None

Parent: <u>Backend Services</u>

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Maryam Agaiby
Resolution:	Done	Votes:	0
Labels:	None		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	KEKO24-85	Create NestJS resources (controller,	Subtask	To Do	
Sprint:					
Rank:	0 i000hb:				

Description

Add more context here (acceptance criteria etc...)

[KEKO24-83] Backend Services Created: 02/Apr/25 Updated: 14/Apr/25 Due: 06/Apr/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Epic	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Start date:	24/Mar/25
Rank:	0 i000h3:

Description

Create feature tasks in Jira under this epic.

Each feature should require necessary API endpoints and database entities etc.

Features:

- Jobs Maryam Agaiby Uzair Irshad
- Reviews Desmond Deveh Abdul Sami
- Resume
- Attachments
- Audit Log <u>Kseniia Kochetova Arkar Myint Myat</u>
 User management (Partially done already) <u>Anastasiia Muhammad Rashid</u>
- Authentication and authorization (Part of User management) Anastasiia Muhammad Rashid

[KEKO24-82] <u>Create Docker Compose File for PostgreSQL Database</u> Created: 31/Mar/25 Updated: 07/Apr/25 Resolved: 07/Apr/25			
Status:	Done		
Project:	Keko 2024		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Desmond Deveh
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 2	
Rank:	0 i000a9:	
Participants:	Arkar Myint Myat, Desmond Deveh	

Description

We need a Docker Compose file that can launch our project's PostgreSQL database in a local environment. This will help streamline the development process and ensure that everyone on the team can easily run and test the database.

Your responsibilities:

- 1. Research Docker Compose configurations for running PostgreSQL in a container.
- 2. Ensure the Compose file includes best practices for persisting data (e.g., volumes).
- 3. Document the required environment variables (e.g., database name, username, password).

Please add all relevant information to our Confluence space so future team members know:

- How to set up the Docker environment.
- How to start and stop PostgreSQL with Docker Compose.
- Any common troubleshooting tips related to Docker/PostgreSQL.

Acceptance Criteria

- A valid Docker Compose file exists in the repository, tested and confirmed to launch PostgreSQL successfully.
- Persistent storage is correctly configured (so data isn't lost upon container restart).

- Confluence documentation is complete and includes:
 - o Steps for running Docker Compose.
 - o Steps for verifying PostgreSQL is up and responding.
 - Troubleshooting or known issues.

Hints/Resources

- Look into official PostgreSQL Docker images.
- Make sure to map ports appropriately so PostgreSQL is accessible.
- Consider how environment variables can be used to store credentials.

Note: Do not commit any private credentials to the repository.

UI Components (KEKO24-35)

[KEKO24-81] Avatar Render Created: 24/Mar/25 Updated: 23/May/25 Resolved: 31/Mar/25

"[KEKU24-81]	Avatar Kender Created: 24/Mar/25 Updated: 23/May/25 Resolved: 31/Mar/25
Status:	Done
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None
Parent:	<u>UI Components</u>

Type:	Task	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 2
Rank:	0 i000b0:

Description

User Story: Develop a Reusable Avatar Component for Website

As a frontend developer,

I want to use a reusable avatar component with configurable inputs, *so that* I can easily display different avatars without redundant code.

Acceptance Criteria:

1. Custom Avatar Rendering:

- o Given the developer provides a custom image source URL,
- When the avatar component is rendered,
- o *Then* it should display the provided image correctly.

2. Avatar Size Variants:

- o Given the developer specifies a size property (e.g., small, medium, large),
- o When the avatar component is rendered,
- Then it should adjust its size accordingly.

3. Placeholder Support:

- o Given the provided avatar source is invalid or unavailable,
- o When the component attempts to render the avatar,
- Then it should fallback to a placeholder image.

4. Accessibility Compliance:

- o Given the component is used in the application,
- o When it is accessed by screen readers,
- o Then it should provide meaningful alt text based on the input.

5. Click Event Handling:

- o Given the developer attaches an event handler to the avatar component,
- When the user clicks on the avatar,
- o *Then* the assigned function should be triggered.

Technical Specifications:

- The component should accept props such as src, alt, size, fallback.
- Implement in React with TypeScript for type safety.
- Unit tests should cover edge cases like missing images.

Wrapper Components (KEKO24-55) [KEKO24-80] Adds for Website Created: 20/Mar/25 Updated: 23/May/25 Resolved: 24/Mar/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium		
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby		
Resolution:	Done	Votes:	0		
Labels:	None	None			
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Sprint:	KEKO24 Sprint 2
Rank:	0 i000gf:
Participants:	Maryam Agaiby, Rizwan Anwar

Description

As a user, I want to

add a reusable wrapper component that contains a main heading, a text body, icon boxes, and one or multiple images, **so that** I can consistently display featured content sections across the application with a standardized layout.

Acceptance Criteria:

Component Structure:

The component must include:

- A main heading.
- A **text body** element that supports rich text or plain text.
- **Four icon boxes** to highlight key features.
- One or multiple **image elements**.

Responsive Design:

• The component must be **responsive**, adjusting its layout for mobile, tablet, and desktop views.

• Icons and images should resize appropriately without losing quality or aspect ratio.

Accessibility:

• All images must include **alt text** for screen readers.

Wrapper Components (KEKO24-55) [KEKO24-79] repeated Created: 19/Mar/25 Updated: 20/Mar/25 Resolved: 20/Mar/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Unassigned
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i000b9:i
Participants:	Maryam Agaiby, Rizwan Anwar

Description

As a user, I want to

add a reusable wrapper component that contains a main heading, a text body, and one or multiple images, **So that**

I can consistently display featured content sections across the application with a standardized layout

Acceptance Criteria:

1. Component Structure:

- The component must include:
 - A main heading element that accepts dynamic text.
 - A text body element that supports rich text or plain text.
 - One or multiple image elements that can be arranged in a gallery or slider layout.

2. Reusability:

- o The component should be designed for reuse across different pages and sections.
- It must accept props to configure the heading text, body content, and image sources.

3. Responsive Design:

- The component must be responsive, adjusting its layout for mobile, tablet, and desktop views.
- Images should resize appropriately without losing quality or aspect ratio.

4. Accessibility:

- o All images must include alt text for screen readers.
- The text elements should use semantic HTML (e.g., <h1>,) for improved accessibility.

Wrapper Components (KEKO24-55)

[KEKO24-78] Password reset page Created: 17/Mar/25 Updated: 31/Mar/25 Resolved: 31/Mar/25

Status: Done

Project: Keko 2024

Components: None

Affects versions: None

Fix versions: None

Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Abdul Sami
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Cloners			
	is cloned by	<u>KEKO24-57</u>	Login page	Done
Sprint:	KEKO24 Sprint 2			
Rank:	0 i000ai:			

Description

User Story: Password Reset Page

Title: Secure Password Reset Process

Description:

As a user, I want to securely reset my password, so that I can regain access to my account without compromising security.

Acceptance Criteria

AC1: Email Verification

Given a user requests a password reset,

When they enter their registered email,

Then they should receive an email with a secure, time-limited reset link.

AC2: Secure Token Validation

Given a user clicks the password reset link,

When the token in the link is expired or invalid,

Then they should see an error message: "This link has expired or is invalid. Please request a new one."

AC3: Strong Password Requirement

Given a user is setting a new password,

When they enter a weak password (e.g., fewer than 8 characters, missing uppercase/lowercase letters, numbers, or special characters),

Then they should see an error message prompting them to use a stronger password.

AC4: Password Update Confirmation

Given a user successfully resets their password,

When the new password is saved,

Then they should receive a confirmation email notifying them of the password change.

AC5: Prevent Reuse of Old Passwords

Given a user is resetting their password,

When they enter a previously used password,

Then they should see an error message: "You cannot use a previously used password. Please choose a new one."

Wrapper Components (KEKO24-55) [KEKO24-77] List wrapper for reviews Created: 17/Mar/25 Updated: 24/Mar/25 Resolved: 24/Mar/25			
Status:	Done		
Project:	Keko 2024		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Parent:	Wrapper Components		

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Desmond Deveh
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 2
Rank:	0 i000ba:i
Participants:	Arkar Myint Myat, Desmond Deveh

Description

User Story: Reviews Listing Wrapper Component on User Profile Page

User Story:

As a visitor, I want to view the reviews of a user on their profile page, so that I can assess their reputation based on feedback from others.

Acceptance Criteria:

1. Review Listing Display

- o Given I visit a user's profile page,
- When the page loads,
- o **Then** I should see a list of reviews including:
 - The reviewer's name
 - The star rating
 - The review comment
 - The date the review was posted

2. Search for a Review

- o Given I am on a user's profile page,
- o When I enter a keyword in the search box,

o **Then** only the reviews that match the keyword should be displayed

3. Filter Reviews

- o Given I am on a user's profile page,
- o When I select a filter option (e.g., highest rating, most recent),
- o Then the review list should update accordingly

4. Accordion (if applicable)

- o Given there are more reviews than can fit on one page,
- o When I navigate to the next or previous page,
- Then I should see the corresponding set of reviews

5. No Reviews Scenario

- o **Given** the user has no reviews,
- o When I visit their profile page,
- o Then I should see a message stating "No reviews available"

Wrapper Components (KEKO24-55)

[KEKO24-76] top-user-data-wrapper Created: 17/Feb/25 Updated: 17/Mar/25 Resolved: 17/Mar/25

Status: Done
Project: Keko 2024

Components: None
Affects versions: None

Fix versions: None

Parent: Wrapper Components

Type:	Story	Priority:	Medium			
Reporter:	Desmond Deveh	Assignee:	Desmond Deveh			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified					
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Relates			
	relates to	<u>KEKO24-63</u>	profile management page	To Do
Sprint:				
Rank:	0 i000fz:			

Description

User Story: Top User Information Component Wrapper

Story Title: Implement a Wrapper Component to Display Logged-in User Information

Story Description:

As a **logged-in user**, I want to see my profile information (profile picture, full name, registration date, location, country) along with action buttons (Add Gigs, Edit Profile) in a dedicated UI component, so that I can quickly access and manage my account details.

Acceptance Criteria:

AC1: Display User Information

Given that the user is logged into the system,

When they access any page with the top-user-information component,

Then the component should display the user's profile picture, full name, registration date, location, and country.

AC2: Profile Picture Handling

Given that the user has a profile picture set,

When the component is loaded,

Then the profile picture should be displayed properly.

And if the user has no profile picture, a default avatar should be shown instead.

AC3: "Add Gigs" Button Functionality

Given that the user is logged in,

When they click on the "Add Gigs" button,

Then they should be redirected to the gig creation modal.

AC4: "Edit Profile" Button Functionality

Given that the user is logged in,

When they click on the "Edit Profile" button,

Then they should be redirected to the profile edit modal.

AC5: Responsive Design

Given that the user is viewing the system on different devices (desktop, tablet, mobile), **When** the component is displayed,

Then it should be fully responsive and adjust its layout accordingly.

Wrapper Components (KEKO24-55) [KEKO24-75] extra Created: 17/Feb/25 Updated: 04/May/25 Resolved: 04/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium		
Reporter:	Maryam Agaiby	Assignee:	Unassigned		
Resolution:	Done	Votes:	0		
Labels:	None				
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Issue links:	Duplicate			
	is duplicated by	KEKO24-62	upload media	To Do
	Relates			
	relates to	<u>KEKO24-66</u>	Footer	Done
	relates to	<u>KEKO24-67</u>	Home page	Done
Sprint:				
Rank:	0 i000fr:			

Wrapper Components (KEKO24-55) [KEKO24-74] Adds for available jobs Created: 17/Feb/25 Updated: 23/May/25 Resolved: 24/Mar/25 Status: Done Project: Keko 2024 Components: None Affects versions: None

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Relates				
	relates to	<u>KEKO24-67</u>	Home page	Done	
Sprint:	KEKO24 Sprint 2				
Rank:	0 i000ba:				
Participants:	Maryam Agaiby,	Rizwan Anwar			

Description

Fix versions:

Parent:

None

Wrapper Components

As a user, I want to

add a reusable wrapper component that contains a main heading, a text body, and one or multiple images, **So that**

I can consistently display featured content sections across the application with a standardized layout

Acceptance Criteria:

1. Component Structure:

- The component must include:
 - A main heading.
 - A text body element that supports rich text or plain text.
 - One or multiple image elements that can be arranged in a gallery or slider layout.

2. Responsive Design:

The component must be responsive, adjusting its layout for mobile, tablet, and desktop views.

 $\circ\quad$ Images should resize appropriately without losing quality or aspect ratio.

3. Accessibility:

- All images must include alt text for screen readers. The text elements should use semantic HTML (e.g., <h1>,) for improved accessibility.

Wrapper Components (KEKO24-73] list-added-jobs/gigs Created: 17/Feb/25 Updated: 23/May/25 Resolved: 15/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium			
Reporter:	Desmond Deveh	Assignee:	Desmond Deveh			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified					
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Relates					
	relates to	<u>KEKO24-63</u>	profile management page	To Do		
Sprint:	KEKO24 S _I	KEKO24 Sprint 3				
Rank:	0 i000i7:					
Participants:	Abdul Sami	, Desmond Devel	n, Maryam Agaiby			

Description

User Story: Post a New Gig or Job as a Client

Title:

As a client, I want to add new gigs or job postings, so that I can find suitable candidates for my work

Acceptance Criteria:

Given the client is logged in and viewing their profile page,

When they click the "Post a Job" button,

Then they should be redirected to the "Create Job" page,

And the page should display a form where they can enter the job details.

AC2: Add a New Job Posting

Given the client is in the create job page,

When they enter valid job details and click "Submit,"

Then the job should be added to their profile's job list and displayed on their profile page.

AC3: Validate Required Fields

Given the client is in the create job page,

When they try to submit without filling in required fields (e.g., job title, description, budget), **Then** they should see an error message indicating the missing information.

AC4: Display Newly Added Jobs in List

Given the client has successfully added a job,

When they return to their dashboard,

Then the new job should be listed.

AC5: Edit an Existing Job Posting

Given the client has existing job postings,

When they click the "Edit" button on a job,

Then the job posting modal should open, allowing them to update the information.

AC6: Delete a Job Posting

Given the client has existing job postings,

When they click the "Delete" button on a job and confirm deletion,

Then the job should be removed from their profile page.

Pages (KEKO24-56)	Pages (KEKO24-56)					
€-[KEKO24-72	[KEKO24-72] extra Created: 17/Feb/25 Updated: 24/Feb/25 Resolved: 24/Feb/25					
Status:	Done					
Project:	<u>Keko 2024</u>					
Components:	None					
Affects versions:	None					
Fix versions:	None					
Parent:	<u>Pages</u>					

Type:	Story	Priority:	Medium		
Reporter:	Maryam Agaiby	Assignee:	Unassigned		
Resolution:	Done	Votes:	0		
Labels:	None				
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Issue links:	Relates			
	relates to	<u>KEKO24-67</u>	Home page	Done
Sprint:				
Rank:	0 i000f3:			

Wrapper Components (KEKO24-71] Category navigation and service cards Created: 17/Feb/25 Updated: 23/Feb/25 Status: To Do Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium		
Reporter:	Maryam Agaiby	Assignee:	Unassigned		
Resolution:	Unresolved	Votes:	0		
Labels:	None				
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Issue links:	Relates			
	relates to	<u>KEKO24-67</u>	Home page	Done
Sprint:				
Rank:	0 i000ev:			

Description

User Story: Service Listing Wrapper with Category Navigation and Service Cards

Title:

Service Listing Wrapper with Navigation Bar and Service Cards

As a user, **I want to** browse services using a navigation bar with categories and view service cards **so that** I can easily find and explore services based on my interests and navigate to the service provider's profile for more information.

Acceptance Criteria:

1. Wrapper Structure and Layout:

• Given a user is viewing the wrapper, when the wrapper contains a navigation bar with main service categories and a grid of service cards, then the user can browse and filter the services easily.

2. Navigation Bar for Filtering Services:

- Given a list of predefined categories in the navigation bar, when a user clicks a category, then the service cards are filtered to show only services belonging to that category.
- Given no category is selected, when the user views the service cards, then a random selection of services from various categories is displayed.
- Given the user interacts with the navigation bar, when a category is hovered or focused on, then visual feedback (like hover and focus effects) is applied for better UX.
- Given the user is using different devices, when interacting with the navigation bar, then
 it adapts to mobile, tablet, and desktop views and is accessible via keyboard and screen
 readers.

3. Service Card Display and Interaction:

- Given a service card is displayed, when the user views it, then the card contains an image representing the service, the categories the service belongs to (displayed as names), the service name, a headline that briefly describes the service, and a "Read More" ghost button that links to the service provider's profile.
- Given a user clicks the "Read More" button, when on a service card, then they are navigated to the service provider's profile page using dynamic routing to load the correct profile.
- Given data props for services, when the wrapper component accepts these props (including category, image, name, and description), then it updates dynamically when a category is selected.

4. General User Experience and Performance Considerations:

- Given no services are available for a selected category, when a user filters the services, then a "No services found" message is displayed.
- Given how the wrapper interacts with users when displaying services, then it should maintain a consistent UI/UX across all interactions with proper spacing and styling to ensure clarity and usability.

Wrapper Components (KEKO24-55) [KEKO24-70] attach-cv-&-letter Created: 17/Feb/25 Updated: 26/Feb/25 Status: To Do

Project: Keko 2024
Components: None
Affects versions: None

Fix versions: None

Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Relates			
	relates to	<u>KEKO24-63</u>	profile management page	To Do
Sprint:				
Rank:	0 i000en:			

Description

Title:

As a profile owner, I want to upload and manage my CV and cover letter, so that I can easily share them when applying for opportunities.

Acceptance Criteria:

AC1: Display Upload Modal

Given the profile owner is on their profile page,

When they click the "Attach CV & Cover Letter" button,

Then a modal popup should appear allowing them to upload or manage their documents.

AC2: Upload CV and Cover Letter

Given the profile owner is in the upload modal,

When they select a valid file (PDF or DOCX) and click "Upload,"

Then the document should be saved and displayed as an attachment in their profile.

AC3: Restrict File Type and Size

Given the profile owner is uploading a document,

When they select a file that is not a PDF or DOCX or exceeds the size limit (e.g., 5MB),

Then they should see an error message indicating the allowed file types and size constraints.

AC4: Replace Existing Documents

Given the profile owner has already uploaded a CV or cover letter,

When they upload a new version,

Then the system should replace the existing file with the new one and update the attachment list.

AC5: Delete Uploaded Documents

Given the profile owner has uploaded a CV or cover letter,

When they click the "Delete" button and confirm deletion,

Then the file should be permanently removed from their profile.

AC6: Preview and Download Documents

Given the profile owner has uploaded a CV or cover letter,

When they click on the file name,

Then they should have the option to preview or download the document.

AC7: Cancel Upload Action

Given the profile owner has opened the upload modal,

When they click "Cancel,"

Then any unsaved changes should be discarded, and the modal should close.

Wrapper Components (KEKO24-55)

[KEKO24-69] Education Created: 17/Feb/25 Updated: 07/May/25 Resolved: 17/Mar/25

Status: Done

Project: Keko 2024

Components: None

Affects versions: None

Fix versions: None

Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Arkar Myint Myat
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Relates			
	relates to	<u>KEKO24-63</u>	profile management page	To Do
Sprint:	KEKO24 Sp	KEKO24 Sprint 2		
Rank:	0 i000az:	0 i000az:		
Participants:	Desmond De	eveh		

Description

Acceptance Criteria:

AC1: Display Education Edit Modal

Given the profile owner is on their profile page,

When they click the "Edit Education" button,

Then a modal popup should appear allowing them to input or update their education details.

AC2: Add New Education Entry

Given the profile owner is in the education edit modal,

When they enter valid educational details and click "Save,"

Then the new education entry should be added to their profile.

AC3: Edit Existing Education Entry

Given the profile owner has existing education entries,

When they modify the details of an entry and click "Save,"

Then the updated details should be reflected on their profile.

AC4: Delete Education Entry

Given the profile owner has existing education entries,

When they click the "Delete" button on an entry and confirm the deletion,

Then the education entry should be removed from their profile.

AC5: Validate Required Fields

Given the profile owner is adding or editing education details,

When they leave a required field (e.g., institution name, degree) empty and attempt to save,

Then they should see an error message indicating that the field is required.

AC6: Cancel Changes

Given the profile owner has opened the education edit modal,

When they click the "Cancel" button,

Then any unsaved changes should be discarded, and the modal should close.

AC7: Save Confirmation

Given the profile owner has successfully saved their education details,

When they return to their profile page,

Then the updated information should be displayed correctly.

	Wrapper Components (KEKO24-55) [KEKO24-68] Search and filter wrapper Created: 17/Feb/25 Updated: 07/May/25 Resolved: 24/Mar/25		
Status:	Done		
Project:	Keko 2024		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Parent:	Wrapper Components		

Type:	Story	Priority:	Medium	
Reporter:	Maryam Agaiby	Assignee:	<u>Uzair Irshad</u>	
Resolution:	Done	Votes:	0	
Labels:	None	None		
Remaining Estimate:	Not Specified			
Time Spent:	Not Specified			
Original estimate:	Not Specified			

Attachments:	image-20250224-081234.png		image-20250224-081215.png	
Issue links:	Relates			
	relates to <u>KEKO24-67</u> Home page		Done	
Sprint:	KEKO24 Spr	KEKO24 Sprint 2		
Rank:	0 i000ae:			
Participants:	Anastasiia, Uzair Irshad			

Description

User Story: Search and Filter Wrapper Component

Title:

Search and Filter Wrapper with Search Bar, Button, and Dropdown Filters

As a User, I want to

search for content using a unified interface that includes a search bar, a search button, and multiple dropde

Acceptance Criteria:

1. Component Structure:

- The component must include:
 - A search bar for entering free-text queries.
 - A search button that initiates the search action.
 - Multiple dropdown filters so that I can refine my search results effectively. These
 - **Location** to find results specific to a chosen area.

- Price Range to filter results within my budget.
- **Seller Level** to sort by seller experience or status.
- **Rating** to prioritize listings with higher customer ratings.
- So that I can easily narrow down my search and find the most relevant results

2. Search Functionality:

- o Users can enter text into the search bar.
- o Pressing the search button or hitting enter triggers the search.

3. Dropdown Filters:

- o Each dropdown should list relevant options.
- o Users can select one or more options from the dropdown filters.
- The component should display the currently selected filter(s) clearly.

4. Reusability and Customization:

- The wrapper should be designed as a reusable component that can be integrated on differen
- o It must accept props to configure:
 - Placeholder text for the search bar.
 - Options for each dropdown filter.

5. Responsive Design:

- The layout should be responsive, ensuring usability across mobile, tablet, and desktop devi
- On smaller screens, the dropdown filters may stack vertically if needed.

6. Accessibility:

- o The search bar and button must be accessible via keyboard and screen readers.
- Dropdown filters should have appropriate ARIA attributes.
- o Ensure that color contrasts and interactive elements meet accessibility standards.

ľ	

For purpose and purpose to purpos	1
In a partie continue, service, and the service of t	
	I
	I
	I
	I
	I
	I
	I
	1
	1
	1
	1
	1
	1
	1
	1
	J

Comments

Comment by Matti Ilvonen [24/Feb/25]

Update Figma design with new design (attached).

Update the acceptance criteria so it explains each filter in detail. What is the data they filter on. When user uses that filter what they should see. Etc. Be specific.

Pages (KEKO24-56)	Pages (KEKO24-56)		
!→[KEKO24-67]	Home page Created: 17/Feb/25 Updated: 23/May/25 Resolved: 11/May/25		
Status:	Done		
Project:	<u>Keko 2024</u>		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Parent:	<u>Pages</u>		

Type:	Story	Priority:	Medium	
Reporter:	Kseniia Kochetova	Assignee:	Desmond Deveh	
Resolution:	Done	Votes:	0	
Labels:	None	None		
Remaining Estimate:	Not Specified	Not Specified		
Time Spent:	Not Specified	Not Specified		
Original estimate:	Not Specified			

Issue links:	Blocks			
	is blocked by	<u>KEKO24-37</u>	Button component	Done
	is blocked by	<u>KEKO24-41</u>	Search-input component	Done
	is blocked by	<u>KEKO24-43</u>	Avatar	Done
	is blocked by	<u>KEKO24-44</u>	Icon component	Done
	is blocked by	<u>KEKO24-46</u>	Profile Picture	Done
	is blocked by	<u>KEKO24-65</u>	Header	Done
	Relates			
	relates to	<u>KEKO24-71</u>	Category navigation and service cards	To Do
	relates to	<u>KEKO24-65</u>	Header	Done
	relates to	<u>KEKO24-66</u>	Footer	Done
	relates to	<u>KEKO24-68</u>	Search and filter wrapper	Done
	relates to	<u>KEKO24-72</u>	extra	Done
	relates to	<u>KEKO24-74</u>	Adds for available jobs	Done
	relates to	<u>KEKO24-75</u>	extra	Done
Sprint:	KEKO24 Spri	nt 3		
Rank:	0 i000in:			
Participants:	Desmond Dev	eh, Maryam A	agaiby	

Description

User Story: Home Page with Header, Footer, Search & Filter, Ads, and Category Navigation

As a user, I want a structured and user-friendly homepage that includes a well-designed header, a functional search and filter system, a section for featured content or ads, a category navigation for browsing services, and a structured footer, **so that** I can easily navigate the website, find relevant services, and access key information efficiently.

Acceptance Criteria:

1. Header with Navigation and User Elements

- The header should include:
 - o A **logo** on the left side.
 - o A navigation bar with links to Home, About, Services, and Contact.
 - o An **avatar** on the right side for logged-in users, which opens a dropdown with "View Profile" and "Log Out" options when clicked.
 - A messages icon next to the avatar, showing a notification badge if there are unread messages.

2. Search & Filter Wrapper

- The search section must contain:
 - o A **search bar** for entering queries.
 - o A **search button** to trigger searches.
 - o **Dropdown filters** (e.g., category, location, date range) for refining results.
 - The component should be **reusable**, allowing different placeholder texts and filter options via props.
 - The UI must be **responsive** and **accessible** with keyboard and screen reader support.

3. Filters for Employers and Freelancers

A filter navigation bar should be available on relevant pages, allowing users to narrow down search results. These filters should apply to both employer job listings and freelancer profiles.

Available Filters:

- **Country** Filter by the user's location.
- **Price Range** Set a minimum and maximum budget or hourly rate.
- **Rating** Filter based on average user ratings.
- **Seller Level** Filter freelancers or employers by experience level (e.g., New, Intermediate, Expert).

Applying any filter should update the displayed results in real-time. If no results match the selected filters, display a **"No results found"** message.

4. Job Listings Section

A dedicated section should be included to display available job postings.

Functionality:

- The section fetches and displays a list of recent or featured **job posts**.
- Each job item should show the following details:

- o Job title
- o **Location** (e.g., country or city)
- o Budget
- If the number of job posts exceeds a predefined limit (e.g., 5 or 10):
 - o A "More Jobs" button should appear below the list.
 - o Clicking the button should navigate the user to the **JobsListPage**, where they can view all available job listings.

5. Featured Content or Ads Section

- A wrapper component should display featured content or advertisements with:
 - o A main heading.
 - o A **text body** (rich or plain text).
 - o **One or multiple images** in a gallery or slider format.
- The section must be **reusable**, accepting dynamic props for text and images.
- The layout should be **responsive** and **accessible**.

5. Footer with Key Information

- The footer should be **divided into four columns**:
 - o Column 1: Company logo and a short description.
 - o Column 2: Links to "About Us," "Services," "Careers," and "Blog."
 - Column 3: Links to "Contact Us," "Privacy Policy," "Terms of Service," and "FAOs."
 - o **Column 4: Clickable social media icons** (Facebook, Twitter, Instagram, LinkedIn).
- Clicking a menu item should **redirect** users to the respective page.
- Clicking a social media icon should **open the platform in a new tab**.
- The layout must be **responsive**, stacking columns vertically on mobile for better readability.

Wrapper Components (KEKO24-55) [KEKO24-66] Footer Created: 17/Feb/25 Updated: 07/May/25 Resolved: 03/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium			
Reporter:	Maryam Agaiby	Assignee:	<u>Uzair Irshad</u>			
Resolution:	Done	Votes:	0			
Labels:	None					
Remaining Estimate:	Not Specified	Not Specified				
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Relates					
	relates to	KEKO24-63	profile management page	To Do		
	relates to	KEKO24-64	Review page	To Do		
	relates to	<u>KEKO24-57</u>	Login page	Done		
	relates to	<u>KEKO24-58</u>	Registration page	Done		
	relates to	<u>KEKO24-59</u>	User profile page	Done		
	relates to	<u>KEKO24-61</u>	Password Reset page	Done		
	relates to	<u>KEKO24-67</u>	Home page	Done		
	relates to	<u>KEKO24-75</u>	extra	Done		
Sprint:	KEKO24 Sprint 2, KEKO24 Sprint 3					
Rank:	0 i000aw:					
Participants:	Anastasiia, U	Jzair Irshad				

Description

User Story: Develop Website Footer with Four Columns

User Story:

As a user, I want a well-structured website footer divided into four columns containing a logo with a short description, two menu item sections, and social media icons, so that I can easily access important links and company information.

Acceptance Criteria:

1. Footer Layout

- o **Given** the user is on any page of the website,
- o When they scroll to the bottom,
- Then they should see a footer divided into four columns.

2. Column 1: Logo and Description

- o **Given** the footer is displayed,
- o When the page loads,
- o **Then** the company logo should be visible in the first column, along with a short company description.

3. Column 2: Menu Items Checklist (Section 1)

- o **Given** the footer is displayed,
- When the user views the second column,
- Then they should see a list of menu items such as "About Us," "Services," "Careers," and "Blog."

4. Column 3: Menu Items Checklist (Section 2)

- o **Given** the footer is displayed,
- o When the user views the third column,
- Then they should see another list of menu items such as "Contact Us," "Privacy Policy," "Terms of Service," and "FAQs."

5. Column 4: Social Media Icons

- o **Given** the footer is displayed,
- o When the user views the fourth column,
- **Then** they should see clickable social media icons (e.g., Facebook, Twitter, Instagram, LinkedIn).

6. Clickable Menu Items

- o **Given** the user is viewing the footer,
- o When they click on a menu item in column 2 or 3,
- o **Then** they should be redirected to the respective page.

7. Clickable Social Media Icons

- o **Given** the user is viewing the footer,
- When they click on a social media icon in column 4,
- Then they should be redirected to the respective social media platform in a new tab.

8. Responsive Design

- o **Given** the user is on a mobile device.
- o **When** they view the footer,
- o **Then** the columns should stack vertically for better readability.

Wrapper Components (KEKO24-55) [KEKO24-65] Header Created: 17/Feb/25 Updated: 23/May/25 Resolved: 03/May/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Maryam Agaiby
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks				
	blocks	<u>KEKO24-67</u>	Home page	Done	
	Relates				
	relates to	<u>KEKO24-63</u>	profile management page	To Do	
	relates to	<u>KEKO24-64</u>	Review page	To Do	
	relates to	<u>KEKO24-57</u>	Login page	Done	
	relates to	<u>KEKO24-58</u>	Registration page	Done	
	relates to	<u>KEKO24-59</u>	User profile page	Done	
	relates to	<u>KEKO24-67</u>	Home page	Done	
Sprint:	KEKO24 Sprint 2, KEKO24 Sprint 3				
Rank:	0 i000dj:				

Description

User Story: Develop Website Header with Logo, Navigation Bar, Avatar, and Messages Icon

User Story:

As a user, I want a well-structured website header so that I can easily navigate the website and access my profile and messages quickly.

Acceptance Criteria:

1. **Logo Display**

- o Given the user is on any page of the website,
- o When the page loads,

o **Then** the company logo should be displayed on the left side of the header.

2. Navigation Bar

- o **Given** the user is on any page,
- o **When** the user views the header,
- o **Then** they should see a navigation bar with links to Home, About, Services, and Contact.

3. Avatar Display

- o Given the user is logged in,
- o When the header loads,
- o **Then** the user's avatar should be displayed on the right side of the header.

4. Messages Icon

- o **Given** the user is logged in,
- o When the header loads,
- o **Then** a messages icon should be visible next to the avatar.

5. Clickable Navigation Elements

- o **Given** the user is viewing the header,
- o When they click on a navigation link,
- Then they should be redirected to the respective page.

6. Avatar Click Opens Profile Options

- o **Given** the user is logged in,
- o When they click on the avatar,
- Then a dropdown should appear with options like "View Profile" and "Log Out."

7. Messages Icon Notification

- o Given the user has unread messages,
- o When they load any page,
- o **Then** a notification badge should appear on the messages icon displaying the unread message count.

Type:	Story	Priority:	Medium			
Reporter:	Desmond Deveh	Assignee:	Abdul Sami			
Resolution:	Unresolved	Votes:	0			
Labels:	None					
Remaining Estimate:	Not Specified	Not Specified				
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Blocks						
	is blocked by	KEKO24-37	Button component	Done			
	is blocked by	<u>KEKO24-38</u>	Input_field_component	Done			
	is blocked by	<u>KEKO24-41</u>	Search-input component	Done			
	is blocked by	<u>KEKO24-42</u>	Rating (Stars) Component	Done			
	is blocked by	KEKO24-47	Modal	Done			
	is blocked by	<u>KEKO24-49</u>	Dropdown	Done			
	Relates						
	relates to	<u>KEKO24-65</u>	Header	Done			
	relates to	<u>KEKO24-66</u>	Footer	Done			
Sprint:	KEKO24 Sprint	KEKO24 Sprint 3					
Rank:	0 i000j3:						
Participants:	Desmond Devel	h					

Description

Acceptance Criteria:

As User(client): I want to leave a review for a freelancer work done.

1. Submit a Review

- o **Given** the user is logged in and on user profile page and has clicked "Leave a review" button,
- When they enter a review and click the "Submit" button,
- o Then the review should be saved and displayed on the review page.

2. View Reviews

- o **Given** the user is on the review page,
- o When they load the page,
- Then all submitted reviews should be displayed in chronological order.

3. Edit Own Review

- o Given the user is logged in and has submitted a review,
- o When they click the "Edit" button next to their review,
- Then they should be able to modify their review and save the changes.

4. Delete Own Review

- o Given the user is logged in and has submitted a review,
- o When they click the "Delete" button,
- o **Then** the review should be removed from the review page.

5. Rating System

- o **Given** the user is submitting a review,
- o When they select a star rating (1-5),
- o **Then** the rating should be displayed alongside their review.

6. Pagination for Reviews

- o **Given** there are more than 10 reviews,
- o When the user scrolls or navigates through the review section,
- o **Then** older reviews should load in batches of 10.

Pages (KEKO24-56)	profile management page Created: 17/Feb/25 Updated: 05/May/25
Status:	To Do
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None
Parent:	<u>Pages</u>

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Arkar Myint Myat
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks			
	is blocked by	<u>KEKO24-62</u>	upload media	To Do
	is blocked by	<u>KEKO24-43</u>	Avatar	Done
	is blocked by	<u>KEKO24-45</u>	Textarea input Field	Done
	is blocked by	<u>KEKO24-46</u>	Profile Picture	Done
	Relates			
	relates to	<u>KEKO24-62</u>	upload media	To Do
	relates to	<u>KEKO24-70</u>	attach-cv-&-letter	To Do
	relates to	<u>KEKO24-65</u>	Header	Done
	relates to	<u>KEKO24-66</u>	Footer	Done
	relates to	<u>KEKO24-69</u>	Education	Done
	relates to	<u>KEKO24-73</u>	list-added-jobs/gigs	Done
	relates to	<u>KEKO24-76</u>	top-user-data-wrapper	Done
Sprint:	KEKO24 Sprint 3			
Rank:	0 i000iv:			
Participants:	Arkar Myint Myat	t, Kseniia Kocheto	ova	

Description

User Stories for Profile Page

1. Media Upload

• **As a user,** I want to upload images, videos, or documents to my profile so that I can showcase my work or portfolio.

- As a user, I want to delete uploaded media so that I can manage my content easily.
- **As a user,** I want to preview uploaded media before saving it so that I can ensure it looks correct.

2. Profile Picture

- **As a user,** I want to set a profile picture separate from my avatar so that I can choose how I appear to others.
- As a user, I want to crop or adjust my profile picture before uploading it so that it fits well.
- As a user, I want to remove my profile picture and use a default placeholder if needed.

3. Text Container (Bio or Description)

- **As a user,** I want to add a short bio or description to my profile so that others can learn about me.
- As a user, I want to edit my bio at any time so that I can keep it up to date.
- **As a user,** I want my bio to have a character limit so that it remains concise and readable.

Wrapper Components (KEKO24-55) [KEKO24-62] upload media Created: 17/Feb/25 Updated: 06/May/25 Status: To Do Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium	
Reporter:	Maryam Agaiby	Assignee:	Unassigned	
Resolution:	Unresolved	Votes:	0	
Labels:	None			
Remaining Estimate:	Not Specified			
Time Spent:	Not Specified			
Original estimate:	Not Specified			

Issue links:	Blocks			
	blocks	KEKO24-63	profile management page	To Do
	blocks	<u>KEKO24-59</u>	User profile page	Done
	is blocked by	KEKO24-53	File Upload	TESTING
	Duplicate			
	duplicates	<u>KEKO24-75</u>	extra	Done
	Relates			
	relates to	<u>KEKO24-63</u>	profile management page	To Do
Sprint:				
Rank:	0 i000ba:9			

Description

User Story 1: Upload Images

As a user, **I want to** upload files (JPEG, PNG, GIF, PDF, DOC), **so that** I can share graphics, designs, written content, reports, or proposals.

Acceptance Criteria:

File Upload (General):

• **Given** I drag and drop a valid file (image or document) into the upload area, **when** the upload is complete, **then** the file should be listed with its name displayed and, for documents, a description field editable.

Image Preview:

• **Given** I upload an image file (JPEG, PNG, GIF), **when** the upload is complete, **then** the image should be displayed in the upload preview.

Error Handling:

• **Given** I upload an unsupported file format, **when** I attempt to upload, **then** I should see an error message indicating the supported formats (JPEG, PNG, GIF, PDF, DOC).

User Story 2: Upload Videos or Add YouTube Links

As a user, **I want to** upload video files (MP4, AVI) or add a YouTube link, **so that** I can showcase animations, video edits, or presentations.

Acceptance Criteria:

Upload Video File:

- **Given** I drag and drop a valid video file into the upload area, **when** the upload is complete, **then** the video should be available for preview as a short clip or thumbnail.
- **Given** I upload a file exceeding the size limit (e.g., 10MB), **when** I attempt to upload, **then** I should see an error message indicating the size restriction.
- **Given** I upload an unsupported video file format, **when** I attempt to upload, **then** I should see an error message indicating the supported formats.

Add YouTube Link:

- **Given** I paste a valid YouTube link into the provided input field, **when** I click the "Add Link" button,
 - **then** the video should be embedded or displayed as a thumbnail preview.
- **Given** I paste an invalid or broken YouTube link, **when** I click the "Add Link" button, **then** I should see an error message indicating that the link is not valid.

User Story 3: Manage Uploaded Files

As a user, I want to manage uploaded files, so that I can delete, add or edit them as needed.

Acceptance Criteria:

- **Given** I have uploaded files, **when** I click the delete icon next to a file, **then** the file should be removed from the upload list.
- **Given** the maximum upload limit (e.g., 5 files) has been reached, **when** I try to upload additional files, **then** I should see an error message indicating the limit.

User Story 4: Add Metadata to Media

As a user, **I want to** add metadata such as a title and description to uploaded files, **so that** I can provide context for each file.

Acceptance Criteria:

- **Given** I upload a file, **when** the upload is complete, **then** I should see fields to add or edit the title and description for the file.
- **Given** I leave the metadata fields blank, **when** I save the upload, **then** I should see a default title generated from the file name.

User Story 5: Media and Profile Management for Admins

As an admin, **I want to** monitor and manage uploaded media files and user profiles, **so that** I can ensure compliance and organization across the platform.

Acceptance Criteria:

Media Management:

- **Given** I am on the admin dashboard, **when** I view the media management section, **then** I should see a list of all uploaded files with their metadata and upload details (e.g., user, date, file type).
- **Given** I identify inappropriate or irrelevant files, **when** I click the "Delete" button for a file, **then** it should be removed from the system.
- **Given** I want to adjust upload limits or allowed file types, **when** I access the settings panel, **then** I should be able to update these parameters and save the changes.
- **Given** a file has missing or incorrect metadata, **when** I click the "Edit" button for that file, **then** I should be able to update the metadata and save the changes.

User Profile Moderation:

- **Given** I am on the admin dashboard, **when** I access a user profile, **then** I should have editing permissions (e.g., update profile details, deactivate account).
- **Given** a user reports an inappropriate profile, **when** I view the reported profiles section, **then** I should see a list of flagged profiles with reasons for reporting.
- **Given** I review a reported profile and find it inappropriate, **when** I click the "Moderate" button, **then** I should have options to warn the user, edit the profile, or deactivate the account.

Pages (KEKO24-56)

[KEKO24-61] Password Reset page Created: 17/Feb/25 Updated: 28/Apr/25 Resolved: 28/Apr/25

Status: Done
Project: Keko 2024
Components: None
Affects versions: None
Fix versions: None

Parent: Pages

Type:	Story	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Abdul Sami
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks	Blocks				
	is blocked by	KEKO24-37	Button component	Done		
	is blocked by	<u>KEKO24-38</u>	Input_field_component	Done		
	is blocked by	<u>KEKO24-39</u>	Logo Component	Done		
	Relates					
	relates to	<u>KEKO24-66</u>	Footer	Done		
Sprint:						
Rank:	0 i000cn:					

Description

User Story: Password Reset Page with Essential UI Components

Title: Password Reset Page with Input Field, Button, and Logo

User Story:

As a user, I want to reset my password because I forgot it.

Acceptance Criteria:

AC1: The page includes a clearly labeled email input field

Given the user is on the password reset page,

When they view the form,

Then they should see an input field labeled "Email Address" where they can enter their registered email.

AC2: The reset button submits the password reset request

Given the user has entered a valid email,

When they click the "Reset Password" button,

Then the system should process the request and trigger a password reset email.

AC3: The button remains disabled until a valid email is entered

Given the user is on the password reset page,

When the email input field is empty or contains an invalid format,

Then the "Reset Password" button should remain disabled.

AC4: The company logo is prominently displayed

Given the user is on the password reset page,

When they look at the top section of the page,

Then they should see the company logo, ensuring brand consistency.

AC5: Proper validation and error messaging

Given the user enters an incorrectly formatted email (e.g., missing '@' or domain), When they try to proceed,

Then an error message should appear prompting them to enter a valid email.

Pages (KEKO24-56) [KEKO24-60] home2 Created: 17/Feb/25 Updated: 24/Feb/25 Resolved: 24/Feb/25			
Status:	Done		
Project:	Keko 2024		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Parent:	Pages		

Type:	Task	Priority:	Medium		
Reporter:	Desmond Deveh	Assignee:	Unassigned		
Resolution:	Done	Votes:	0		
Labels:	None	None			
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Sprint:	
Rank:	0 i000cf:

Description

User Story: Home Page with Header, Footer, Search & Filter, Ads, and Category Navigation

As a user, I want a structured and user-friendly homepage that includes a well-designed header, a functional search and filter system, a section for featured content or ads, a category navigation for browsing services, and a structured footer, **so that** I can easily navigate the website, find relevant services, and access key information efficiently.

Acceptance Criteria:

1. Header with Navigation and User Elements

- The header should include:
 - o A **logo** on the left side.
 - o A navigation bar with links to Home, About, Services, and Contact.
 - o An **avatar** on the right side for logged-in users, which opens a dropdown with "View Profile" and "Log Out" options when clicked.
 - o A **messages icon** next to the avatar, showing a **notification badge** if there are unread messages.

2. Search & Filter Wrapper

• The search section must contain:

- o A **search bar** for entering queries.
- o A **search button** to trigger searches.
- o **Dropdown filters** (e.g., category, location, date range) for refining results.
- The component should be **reusable**, allowing different placeholder texts and filter options via props.
- The UI must be **responsive** and **accessible** with keyboard and screen reader support.

3. Category Navigation & Service Listings

- A category-based navigation bar should be present, allowing users to browse services by category.
- Clicking a category should **filter service listings** to display only services in that category.
- Service cards should include:
 - o An **image** representing the service.
 - o A category label.
 - o The **service name** and a short description.
 - o A "**Read More**" button that navigates to the provider's profile.
- If no services match the selected category, a "No services found" message should appear.

4. Featured Content or Ads Section

- A wrapper component should display featured content or advertisements with:
 - o A main heading.
 - o A **text body** (rich or plain text).
 - o **One or multiple images** in a gallery or slider format.
- The section must be **reusable**, accepting dynamic props for text and images.
- The layout should be **responsive** and **accessible**.

5. Footer with Key Information

- The footer should be **divided into four columns**:
 - o Column 1: Company logo and a short description.
 - o Column 2: Links to "About Us," "Services," "Careers," and "Blog."
 - Column 3: Links to "Contact Us," "Privacy Policy," "Terms of Service," and "FAQs."
 - Column 4: Clickable social media icons (Facebook, Twitter, Instagram, LinkedIn).
- Clicking a menu item should **redirect** users to the respective page.
- Clicking a social media icon should **open the platform in a new tab**.
- The layout must be **responsive**, stacking columns vertically on mobile for better readability.

Pages (KEKO24-56) Liser profile page Created: 17/Feb/25 Updated: 07/May/25 Resolved: 14/Apr/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Pages

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Desmond Deveh
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks				
	is blocked by	KEKO24-36	Avatar	To Do	
	is blocked by	KEKO24-62	upload media	To Do	
	is blocked by	<u>KEKO24-37</u>	Button component	Done	
	is blocked by	<u>KEKO24-42</u>	Rating (Stars) Component	Done	
	is blocked by	<u>KEKO24-49</u>	Dropdown	Done	
	Relates				
	relates to	<u>KEKO24-65</u>	Header	Done	
	relates to	<u>KEKO24-66</u>	Footer	Done	
Sprint:	KEKO24 Sprint 2, KEKO24 Sprint 3				
Rank:	0 i000au:				
Participants:	Arkar Myint Myat, Desmond Deveh				

Description

User Stories for Profile Page Display

1. Viewing the Profile

- As a visitor, I want to view a user's profile page so that I can learn more about them.
- **As a visitor,** I want to see the user's avatar and profile picture so that I can recognize them.
- **As a visitor,** I want to see the user's bio so that I can understand their background or expertise.

• **As a visitor,** I want to see the media uploads on the profile so that I can review their work or portfolio.

2. Reviews and Star Ratings

- **As a visitor,** I want to give a star rating to a profile so that I can provide feedback on my experience.
- **As a visitor,** I want to submit a written review along with my star rating so that I can share detailed feedback.
- **As a visitor,** I want to see the average star rating of a profile so that I can quickly assess its reputation.
- **As a visitor,** I want to see all reviews left by other users so that I can make an informed judgment about the profile.

3. Comments on Profile

- **As a visitor,** I want to leave a public comment on a profile so that I can share my thoughts or ask questions.
- **As a visitor,** I want to edit or delete my comment so that I can correct mistakes or remove unwanted messages.
- **As a visitor,** I want to report inappropriate comments so that harmful content can be addressed.

4. Profile Owner Actions

- **As a profile owner,** I want to reply to comments on my profile so that I can engage with visitors.
- **As a profile owner,** I want to report inappropriate comments on my profile so that harmful content can be reviewed and removed.
- **As a profile owner,** I want to see who left a review so that I can understand feedback better.

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Matti Ilvonen
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks	Blocks				
	is blocked by	<u>KEKO24-37</u>	Button component	Done		
	is blocked by	<u>KEKO24-38</u>	Input_field_component	Done		
	is blocked by	<u>KEKO24-39</u>	Logo Component	Done		
	Relates					
	relates to	<u>KEKO24-65</u>	Header	Done		
	relates to	<u>KEKO24-66</u>	Footer	Done		
Sprint:	KEKO24 Sprint	2				
Rank:	0 i000ak:					
Participants:	Kseniia Kochetova, Syed Muneer Ahmed					

Description

• User Story:

As a new user, I want to register my email and password, so that I can create an account.

• Acceptance Criteria:

- Given I am on the registration page, when I provide a valid email, password, and confirm the password, then I should proceed to the next step of the registration process.
- Given I am on the registration page, when I enter weak password, then I should see an error message prompting me valid parameters I need for password (length, etc).
- o **Given** I enter mismatched passwords, **when** I submit the form, **then** I should see an error message prompting me to re-enter matching passwords.
- o **Given** any required field is left empty, **when** I click the Register button, **then** I should see an error message prompting me to fill it.

o **Given** I provide an already-registered email, **when** I submit the form, **then** I should see an error message saying, "Email already registered."

Pages (KEKO24-56) [KEKO24-57] Login page Created: 17/Feb/25 Updated: 07/May/25 Resolved: 17/Mar/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Pages

Type:	Story	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Muhammad Rashid
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks				
	is blocked by	<u>KEKO24-37</u>	Button component	Done	
	is blocked by	<u>KEKO24-38</u>	Input_field_component	Done	
	is blocked by	<u>KEKO24-39</u>	Logo Component	Done	
	Cloners				
	clones	<u>KEKO24-78</u>	Password reset page	Done	
	Relates				
	relates to	<u>KEKO24-65</u>	Header	Done	
	relates to	<u>KEKO24-66</u>	Footer	Done	
Sprint:	KEKO24 Sprint 2				
Rank:	0 i000b2:				
Participants:	Abdul Sami, Muhammad Rashid				

Description

User Storybook: Login Page

User Story 1: User Login

Title: User Authentication via Login Page

Description:

As a user, I want to log in to my account using my email and password, so that I can access my personalized dashboard.

Acceptance Criteria:

• AC1: Successful Login

- o Given the user has a registered account,
- o When they enter a valid email and password and click "Login",
- o Then they should be redirected to the dashboard.

• AC2: Invalid Credentials Handling

- o Given the user enters incorrect login credentials,
- o When they attempt to log in,
- o Then they should receive an error message: "Invalid email or password."

• AC3: Remember Me Functionality

- o Given the user selects the "Remember Me" checkbox,
- o When they log in successfully,
- o Then their session should persist even after closing and reopening the browser.

User Story 3: Social Login

Title: Google & GitHub Authentication

Description:

As a user, I want to log in using my Google or GitHub account, so that I don't have to remember another password.

Acceptance Criteria:

• AC1: Google Login

- o Given the user clicks "Sign in with Google",
- o When they authenticate successfully via Google,
- o Then they should be logged into their account and redirected to the dashboard.

• AC2: GitHub Login

- o Given the user clicks "Sign in with GitHub",
- o When they authenticate successfully via GitHub,
- o Then they should be logged into their account and redirected to the dashboard.

• AC3: Failed Third-Party Authentication Handling

- o Given the user tries to sign in with Google or GitHub,
- When authentication fails,
- Then they should receive an error message: "Third-party authentication failed. Please try again."

User Story 4: Security Features

Title: Secure Authentication Mechanism

Description:

As a developer, I want to ensure that the login page is secure, so that user credentials and sessions are protected.

Acceptance Criteria:

• AC1: Password Encryption

- o Given the user enters their password,
- o When it is sent to the server,
- o Then it should be securely encrypted using hashing before storage.

• AC2: Rate Limiting

- o Given a user attempts multiple failed logins,
- o When they exceed 5 attempts within 5 minutes,
- Then they should be temporarily locked out with a message: "Too many failed attempts. Try again later."

• AC3: Session Expiry

- o Given a user is logged in,
- When they are inactive for 30 minutes,
- o Then they should be automatically logged out and redirected to the login page.

[KEKO24-56] Pages Created: 17/Feb/25 Updated: 17/Feb/25 Due: 11/Apr/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Epic	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Start date:	12/Mar/25
Rank:	0 i000bb:

[KEKO24-55] Wrapper Components Created: 17/Feb/25 Updated: 07/May/25 Due: 17/Mar/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Epic	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Start date:	18/Feb/25
Rank:	0 i000bj:

UI Components (KEKO24-35) [KEKO24-54] Drag and drop Created: 10/Feb/25 Updated: 14/Apr/25 TESTING

Project: Keko 2024 **Components:** None **Affects versions:** None Fix versions: None

Status:

UI Components Parent:

Type:	Task	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Syed Muneer Ahmed
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 1, KEKO24 Sprint 2, KEKO24 Sprint 3
Rank:	0 i000an:

Description

As a user,

I want to be able to drag and drop files into a designated area on the screen

I can quickly and easily upload files without navigating a file picker.

Acceptance Criteria

- 1. There is a visible drop zone or area labeled clearly for dragging files.
- 2. When a file is dragged over the drop zone, it highlights or indicates that a file can be dropped.
- 3. Once I drop my file into the zone, it starts the upload process or immediately shows that the file is accepted.
- 4. If the file is accepted, the application shows a success message or displays the file's name. If the file type is not supported, the application displays an error or helpful message.

UI Components (KEKO24-35)

[KEKO24-53] File Upload Created: 10/Feb/25 Updated: 14/Apr/25

[HEHOZ. CO.	THE OPTOWN Cleated. 19/1/09/25 Optimed. 14/1/19/1/25
Status:	TESTING
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None
Parent:	<u>UI Components</u>

Type:	Task	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Syed Muneer Ahmed
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks					
	blocks	<u>KEKO24-62</u>	upload media	To Do		
Sprint:	KEKO24	KEKO24 Sprint 1, KEKO24 Sprint 2, KEKO24 Sprint 3				
Rank:	0 i000at:					

Description

As a user,

I want to see a simple button that lets me upload a file from my computer so that

I can share that file in the application.

Acceptance Criteria

- 1. There is a clearly visible button that indicates I can upload a file.
- 2. When I click the button, it allows me to select a file from my device.
- 3. Once I choose a file, the component clearly shows that a file has been selected (for example, by displaying the file name).
- 4. The application receives the file information so it can be processed, stored, or sent to another service.

UI Components		(25 V. I.) (2724 - 25 D. I.)	045105			
Status:	Done Created: 20/Jan	1/25 Updated: 07/May/25 Resolved:	24/Feb/25			
Project:	Keko 2024					
Components:	None					
Affects versions:	None	None				
Fix versions:	None	None				
Parent:	UI Components					
Type:	Task	Priority:	Medium			
Reporter:	Anastasiia	A ssignee	Anastasiia			

Type:	Task	Priority:	Medium			
Reporter:	<u>Anastasiia</u>	Assignee:	<u>Anastasiia</u>			
Resolution:	Done	Done Votes: 0				
Labels:	None	None				
Remaining Estimate:	Not Specified	Not Specified				
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Attachments:	image-20250506-135634.png image-20250506-135854.png image-20250506-141241.png	
Sprint:	KEKO24 Sprint 1	
Rank:	0 i0006z:	

Description

A Badge is a small UI element used to display brief, contextual information such as status, category, or count.
4 types of badge:
To a more religious natural parties, beautiful parties beautiful to district, representative as manifestable for the parties of the parties o

To paperus milipanes, seniemad poposa, Brimania, escri delle lan represione, representate servizione, Thaprimo, voi zionne yazonere ne representati delle a represi parenquese.	1
To a part interes more of prices forces, are fell for represe, represent engine. Higher, or some variety or personal pair a specie paragraph.	
	,
To particip indigene, sectional propose, formace, and daily has represent, indigeneously are spice, This for, in Eurito years of Section of Section stocks (Personal Section).	

UI Components (KEKO24-35) [KEKO24-51] Date Picker Created: 20/Jan/25 Updated: 07/May/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None

Fix versions: None
Parent: UI Components

Affects versions:

None

Type:	Task	Priority:	Medium			
Reporter:	Uzair Irshad	Assignee:	Uzair Irshad			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified					
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Sprint:	KEKO24 Sprint 1
Rank:	0 i0008i:9

Description

ask Description:

Develop a reusable **Date Picker Component** leveraging Radix UI. This component should enable developers to integrate it into different parts of the application where date selection is required. It should allow the user to input or select a date and support accessibility, ease of use, and flexibility.

Sub-Tasks:

1. Define Component Requirements

- Specify the input props:
 - selectedDate: To define a pre-selected date.
 - onDateChange: Callback for returning the selected date.
 - minDate and maxDate: To restrict date selection range.
 - placeholder: Custom placeholder text for the date input field.
- o Support both manual text input and calendar-based selection.

2. Implement with Radix UI

- o Utilize Radix UI primitives (Popover, Dialog, etc.) to create the dropdown calendar.
- Implement functionality for:
 - Navigating between months.

- Highlighting today's date.
- Showing disabled dates based on minDate and maxDate.

3. Design and Styling

- Style the component to match the design system, ensuring a clean and intuitive user interface.
- o Include clear indications for invalid input or out-of-range selections.

4. Validation and Error Handling

- o Validate the user-input date format (e.g., YYYY-MM-DD or MM/DD/YYYY).
- o Provide error feedback for invalid dates or dates outside the allowed range.

5. Accessibility

- Ensure the component supports keyboard navigation:
 - Arrow keys to move between dates.
 - Enter key to select a date.
- Add ARIA attributes for accessibility compliance.

6. Localization

- o Include support for multiple locales, allowing formats such as:
 - MM/DD/YYYY (US format).
 - DD/MM/YYYY (EU format).

7. **Testing**

- o Write unit tests for:
 - Correct rendering of the calendar and input field.
 - Validation for date input.
 - Date range restrictions.
- o Create snapshot tests for visual regression.
- Add integration tests for key interactions (e.g., date selection via calendar and manual input).

8. **Documentation**

- Write a comprehensive README or developer documentation:
 - API details (props, events, etc.).
 - Usage examples for form integrations.
 - Customization and configuration options.
- o Include examples for:
 - Basic usage with a default date.
 - Using minDate and maxDate.
 - Localized formats.

Acceptance Criteria:

AC1: Date Selection via Calendar

- **Given** a date picker is rendered,
- When the user opens the calendar and selects a date,
- Then the selected date should appear in the input field.

AC2: Manual Date Input

- **Given** a date picker is rendered,
- When the user manually inputs a valid date,

• Then the input field should reflect the entered date, and the calendar should update accordingly.

AC3: Date Range Validation

- Given a minDate and maxDate are provided,
- When the user attempts to select or input a date outside this range,
- Then the component should display an error and prevent selection.

AC4: Accessibility

- Given the date picker is rendered,
- When the user navigates using the keyboard or a screen reader,
- Then the component should support full navigation and provide ARIA-compliant feedback.

AC5: Localization Support

- Given the date picker is configured with a specific locale,
- When the user interacts with it,
- Then the input field and calendar should adhere to the specified date format.

AC6: Clear and Reset Functionality

- Given a date has been selected,
- When the user clicks a clear button (if provided),
- Then the date input should be reset to its initial state.

UI Components (KEKO24-35) [KEKO24-49] Dropdown Created: 20/Jan/25 Updated: 19/May/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None

Type:	Task	Priority:	Medium			
Reporter:	Maryam Agaiby	Assignee:	Maryam Agaiby			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified					
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Blocks	Blocks				
	blocks	<u>KEKO24-64</u>	Review page	To Do		
	blocks	<u>KEKO24-59</u>	User profile page	Done		
Sprint:	KEKO24	KEKO24 Sprint 1				
Rank:	0 i00072:) i00072:				

Description

Parent:

User Story: Reusable Dropdown Component

UI Components

Title:

As a developer, I want a reusable and extensible Dropdown component so that I can integrate dynamic dropdown menus into my applications with ease, ensuring customization, accessibility, and robust functionality.

Description:

Create a reusable and extensible Dropdown component within the component library. This component will serve as a wrapper for various dropdown menus, allowing developers to pass specific options as input properties. It should support customization options such as size, style, and accessibility features

Features

- Can be controlled or uncontrolled.
- Supports items, labels, and groups of items.
- Customize side.
- Optionally render a pointing arrow.
- Focus is fully managed.
- keyboard navigation.

Example of how the Dropdown component would be used in code:

The dropdown component must include:

- 1. **Dynamic Display:** Accept a prop (e.g., options) to dynamically load and display the appropriate dropdown options.
- 2. **Customization:** Support size, style, and additional styling via props.
- 3. **Testing:** Unit tests to validate component functionality.
- 4. **Documentation:** Add usage examples and edge cases to Storybook.

Acceptance Criteria:

1. Dynamic Dropdown Display:

- o Given valid options are provided,
- When the Dropdown component is displayed,
- o Then the correct dropdown options are shown.

2. Styling and Customization:

- o Given size, style, and additional styles are passed as props,
- o When the Dropdown component is displayed,
- o Then the styles are correctly applied to the dropdown.

3. Accessibility:

o keyboard navigation.

4. Storybook Documentation:

- o Given the Storybook is accessed,
- o When the Dropdown component is previewed,
- Then examples of different dropdown options, sizes, and styles are available for users.

Technical Tasks:

- 1. Create the Dropdown component with a prop (options) for dynamic option selection.
- 2. Implement prop validation to ensure valid inputs for options, size, and style.
- 3. Add functions to manage dropdown options and styles.
- 4. Write unit tests for:
 - o Valid option display.
 - o Default behavior for invalid or missing options.
 - o Style and accessibility validations.
- 5. Add stories in Storybook showcasing:
 - o Examples of multiple dropdown options.
 - o Prop usage for size, style.

Definition of Done:

- 1. All acceptance criteria are met.
- 2. Code is reviewed and approved.
- 3. Unit tests achieve at least 90% coverage.
- 4. Component passes accessibility checks (e.g., with tools like axe-core).
- 5. Stories are added to Storybook with clear examples and documentation.
- 6. The component is merged to the main branch with no conflicts and is ready for release.

UI Components (KEKO24-35) [KEKO24-47] Modal Created: 20/Jan/25 Updated: 17/Mar/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: UI Components

Type:	Task	Priority:	Medium			
Reporter:	Arkar Myint Myat	Assignee:	Arkar Myint Myat			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified					
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Blocks			
	blocks	<u>KEKO24-64</u>	Review page	To Do
Sprint:	KEKO24 Sprint 1			
Rank:	0 i00093:			

Description

Create a Reusable and Extensible Modal Component for the Component Library

Description

Develop a reusable and extensible Modal component that acts as a wrapper for various modal types. This component will allow developers to pass specific configurations (e.g., title, content, footer) as input properties. It should support customization options like size, theme, animations, and accessibility features (e.g., focus trapping, ARIA labels).

Example Usage:

```
import Modal from './Modal';

<Modal
  isOpen={true}
  onClose={() => console.log('Modal closed')}
  title="Example Modal"
  size="medium"
  ariaLabel="Example Modal"
```

```
>
  This is the modal content.
  <div slot="footer">
        <button onClick={() => console.log('Confirmed')}>Confirm</button>
        </div>
  </Modal>
```

The Modal component must include:

- **Dynamic Rendering**: Render modal content dynamically based on the provided props.
- **Customization**: Support props for size, theme, animations, and additional styles.
- Accessibility: Implement ARIA attributes, focus management, and keyboard navigation.
- Error Handling: Provide fallback behavior when required props are missing.
- **Testing**: Include unit tests to validate functionality and edge cases.
- **Documentation**: Add usage examples and edge cases to Storybook.

Acceptance Criteria

Dynamic Modal Rendering

- Given the isOpen prop is true,
- When the Modal component is rendered,
- Then the modal is displayed with the specified title and content.

Styling and Customization

- **Given** size, theme, or additional styles are passed as props,
- When the Modal component is rendered,
- Then the styles and themes are correctly applied.

Accessibility

- Given ARIA attributes (e.g., ariaLabel) are provided via props,
- When the Modal component is rendered,
- Then screen readers correctly announce the modal, and keyboard focus is trapped within it.

Closing Behavior

- Given the onclose callback is provided,
- When the close button is clicked, or the overlay is clicked,
- Then the modal is closed, and the onclose callback is triggered.

Error Handling

- Given an invalid or missing isopen prop is provided,
- When the Modal component is rendered,
- Then a default behavior (e.g., modal hidden) is implemented with a console warning.

Storybook Documentation

- **Given** the Storybook is accessed,
- When the Modal component is previewed,
- Then examples of different modal sizes, themes, and behaviors are available.

Technical Tasks

1. Component Implementation:

- o Create the Modal component with props for:
 - isopen (boolean): Controls visibility.
 - title (string): Title of the modal.
 - onClose (function): Callback for closing the modal.
 - size (string): Determines modal size (e.g., small, medium, large).
 - theme (string): Applies pre-defined styling themes.
 - ariaLabel (string): Accessibility label for the modal.
 - animations (boolean): Toggles modal animations.

2. Focus Management:

- o Implement focus trapping within the modal when it is open.
- o Ensure focus returns to the triggering element when the modal closes.

3. Prop Validation:

- o Validate props to ensure:
 - isopen is a boolean.
 - size matches allowed values.
 - ariaLabel is provided and non-empty.

4. Error Handling:

 Provide a fallback for missing or invalid props (e.g., hide modal and log a warning).

5. Unit Tests:

- Write unit tests for:
 - Rendering the modal based on isopen.
 - Handling of the onClose callback.
 - Validation of styles and themes.
 - Accessibility behavior (focus trapping, ARIA attributes).
 - Edge cases (e.g., missing props).

6. Storybook Documentation:

- Add stories showcasing:
 - Modal with a simple title and content.
 - Different sizes, themes, and animations.
 - Custom footers and dynamic content.
 - Edge cases (e.g., invalid or missing props).

Definition of Done

- All acceptance criteria are met.
- Code is reviewed and approved.
- Unit tests achieve at least 90% coverage.
- Component passes accessibility checks (e.g., focus trapping, ARIA attributes).
- Stories are added to Storybook with clear usage examples and edge cases.

• The corelease	mponent is merg	ged into the m	ain branch w	ith no confli	cts and is rea	The component is merged into the main branch with no conflicts and is ready for release.		

UI Components (KEKO24-35)

[KEKO24-46] Profile Picture Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25

Profile Picture Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25
Done
<u>Keko 2024</u>
None
None
None
<u>UI Components</u>

Type:	Task	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Abdul Sami
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks				
	blocks	KEKO24-63	profile management page	To Do	
	blocks	<u>KEKO24-67</u>	Home page	Done	
Sprint:	KEKO24 Sprint 1				
Rank:	0 i000a5:				

Description

User Story: Develop a Profile Picture Component

As a frontend developer,

I want to use a reusable profile picture component with customizable options, so that I can display user profile images in different sections of the website.

Acceptance Criteria:

1. Default Profile Picture Rendering:

- o Given the developer does not provide a profile picture source,
- o When the profile picture component is rendered,
- o *Then* it should display a default profile picture image.

2. Custom Profile Picture Rendering:

- o Given the developer provides a valid profile picture URL,
- o When the profile picture component is rendered,
- o *Then* it should display the provided image correctly.

3. Editable Profile Picture:

- o Given the editable prop is set to true,
- o When the user clicks on the profile picture,

o Then an upload dialog should appear allowing the user to change the image.

4. Placeholder Support for Profile Picture:

- o Given the provided profile picture URL is invalid or unavailable,
- o When the profile picture component is rendered,
- Then it should show a default placeholder profile picture.

5. Profile Picture Size Options:

- o Given the developer specifies a size (small, medium, large),
- o When the profile picture component is rendered,
- o *Then* it should adjust its size accordingly.

6. Accessibility Compliance:

- o Given the component is used in the application,
- When it is accessed by screen readers,
- o Then it should provide meaningful alt text such as "User profile picture."

7. Custom Styles:

- Given the developer provides custom styles or class names,
- o When the profile picture component is rendered,
- o *Then* it should apply the provided styles correctly.

Technical Specifications:

- Props: src, alt, size, className, editable, onClick.
- Should include an upload feature if editable.
- Implement using React and TypeScript.
- Maintain consistent styling with other UI components.
- Write unit tests for different usage cases.

UI Components (KEKO24-45] Textarea input Field Created: 20/Jan/25 Updated: 07/May/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: UI Components

Type:	Task	Priority:	Medium		
Reporter:	Anastasiia	Assignee:	Anastasiia		
Resolution:	Done	Votes:	0		
Labels:	None				
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Issue links:	Blocks
	blocks <u>KEKO24-63</u> profile management page To Do
Sprint:	KEKO24 Sprint 1
Rank:	0 i00075:

Description

Here are some user stories for developing a Textarea Input Field component in RadixUI:

1. As a user, I want to enter multi-line text in a text area, so that I can provide detailed input.

- Acceptance Criteria:
 - o The textarea should support multiple lines of input.
 - Users can type beyond the visible height of the textarea, and the text area should scroll vertically.
 - o The textarea should expand in height as more text is entered, if necessary.

2. As a user, I want to see placeholder text when the textarea is empty, so that I know what kind of information is expected.

- Acceptance Criteria:
 - o A placeholder text should be visible when the textarea is empty.
 - o Placeholder text should disappear when the user starts typing.

3. As a user, I want the textarea to resize vertically according to the content I enter, so that I don't need to manually scroll or resize it.

- Acceptance Criteria:
 - o The textarea should automatically resize vertically to fit the content.
 - o A maximum height should be enforced, after which vertical scrolling is enabled.
- 4. As a user, I want to be able to copy and paste text into the textarea, so that I can easily insert text from other sources.
 - Acceptance Criteria:
 - o Users should be able to paste text into the textarea without issues.
 - o Pasting large amounts of text should work seamlessly.
- 5. As a user, I want to be able to see my input clearly with proper text formatting, so that I can review and modify my content easily.
 - Acceptance Criteria:
 - o Text should be rendered with proper line breaks and spacing.
 - o Text input should be legible, with consistent font and size.
- 6. As a user, I want to be able to resize the textarea manually if needed, so that I can adjust the field to my preferred size.
 - Acceptance Criteria:
 - The textarea should have a handle at the bottom-right corner to allow manual resizing.
 - Resizing should maintain the same functionality for scrolling and content display.
- 7. As a user, I want to have proper accessibility features, such as a label and focus indication, so that I can interact with the textarea comfortably using assistive technologies.
 - Acceptance Criteria:
 - o The textarea should be associated with a descriptive label.
 - The textarea should be focusable, and focus should be indicated with a visible outline or border.
 - The textarea should support screen readers and provide proper ARIA roles and labels.
- 8. As a developer, I want to be able to set an initial value in the textarea, so that I can pre-fill it with data when required.
 - Acceptance Criteria:
 - o The textarea should accept an initial value through a value prop.
 - o If a value is provided, it should be displayed as the default content of the textarea.

9. As a user, I want to have the option to disable the textarea, so that I cannot interact with it when it's not available for input.

• Acceptance Criteria:

- o The textarea should accept a disabled prop.
- When disabled, the textarea should not be editable or focusable, and it should appear visually dimmed or greyed out.

10. As a user, I want to ensure that my input is limited by a character count, so that I don't exceed a certain length.

• Acceptance Criteria:

- o The textarea should support a maxLength prop to limit the number of characters.
- o A visual indication (e.g., character count) should be displayed to show how many characters have been used and how many are left.

11. As a user, I want the textarea to trigger validation messages if my input is invalid, so that I can correct mistakes.

• Acceptance Criteria:

- o The textarea should trigger validation events (e.g., onInvalid).
- o If the input doesn't meet certain criteria (e.g., required or max length), a validation error message should be displayed near the textarea.

12. As a developer, I want the ability to integrate the textarea with form submission, so that it can be used as a regular form input.

• Acceptance Criteria:

- o The textarea should work seamlessly as part of a form and be included when the form is submitted.
- o It should support typical form behaviors, like validation, and clearing the input on form reset.

Installation of Dependencies

1. Install the necessary libraries:

```
npm install @radix-ui/react-label @radix-ui/react-tooltip
```

Code Example

```
import React, { useState } from 'react';
import * as Label from '@radix-ui/react-label';
import * as Tooltip from '@radix-ui/react-tooltip';
import './App.css'; // You can add your own styles here

const TextareaInputField = ({ label, placeholder, maxLength, required, error }) => {
  const [value, setValue] = useState('');
  const remainingChars = maxLength - value.length;

// Handler for text change
  const handleChange = (event) => {
```

```
setValue(event.target.value);
  };
  return (
    <div className="textarea-container">
      <Label.Root htmlFor="textarea" className="textarea-label">
        {label}
      </Label.Root>
      <textarea
        id="textarea"
        value={value}
        onChange={handleChange}
        placeholder={placeholder}
        maxLength={maxLength}
        required={required}
        className={`textarea ${error ? 'textarea-error' : ''}`}
      {maxLength && (
        <div className="char-count">
          {remainingChars} characters left
        </div>
      ) }
      {error && (
        <Tooltip.Provider>
          <Tooltip.Root>
            <Tooltip.Trigger className="tooltip-trigger">
              <span className="error-message">{error}</span>
            </Tooltip.Trigger>
          </Tooltip.Root>
        </Tooltip.Provider>
    </div>
 );
};
const App = () \Rightarrow {
 return (
    <div className="app">
      <h1>Textarea Input Field Example</h1>
      <TextareaInputField
        label="Your message"
        placeholder="Write something..."
        maxLength={200}
        required={true}
        error="Please write a valid message."
      />
    </div>
 );
} ;
export default App;
```

Explanation:

1. Using Radix UI Label:

o Label.Root is used to associate a label with the <textarea> element, improving accessibility.

o The htmlFor attribute links the label to the id of the textarea, which is important for accessibility.

2. Textarea Element (<textarea>):

- o The textarea value is controlled using React's useState.
- o The maxLength attribute limits the length of the input.
- o Error handling is added by conditionally applying the textarea-error class if an error message is provided.

3. **Tooltip for Errors**:

o If an error message is passed, it is displayed using Radix UI's Tooltip component.

4. Character Counter:

o Shows how many characters are left when the maxLength property is set.

Explanation of Functionality:

- The **Textarea Input** allows for multi-line text input.
- The Character counter shows how many characters are left based on the maxLength value.
- If there is an **error**, it is displayed using Radix UI's Tooltip component, along with an error message.

UI Components (KEKO24-44] Icon component Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: UI Components

Type:	Task	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Matti Ilvonen
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	sue links: Blocks			
	blocks	<u>KEKO24-67</u>	Home page	Done
Sprint:	KEKO24 Sprint 1			
Rank:	0 i0006q:			

Description

Description: Create a reusable and extensible Icon component within the component library. This component will serve as a wrapper for various SVG icons (Font awesome icons), allowing developers to pass a specific icon as an input property. It should support customization options such as size, color, and accessibility features (e.g., ARIA labels).

```
import { faHome } from '@fortawesome/free-solid-svg-icons';
import Icon from './Icon';

/* Example how Icon component would be used in code */
<Icon iconName={faHome} ariaLabel="Home" />
```

The Icon component must include:

- 1. **Dynamic Rendering**: Accept a prop (e.g., iconName) to dynamically load and display the appropriate SVG icon.
- 2. **Customization**: Support size, color, and additional styling via props.
- 3. **Accessibility**: Include ARIA attributes for better screen reader support.
- 4. **Testing**: Unit tests to validate component functionality, including edge cases and props.
- 5. **Documentation**: Add usage examples and edge cases to Storybook.

Acceptance Criteria:

1. Dynamic Icon Loading:

- o Given a valid iconName is provided,
- o When the Icon component is rendered,
- Then the correct SVG icon is displayed.

2. Styling and Customization:

- o Given size, color, and additional styles are passed as props,
- o When the Icon component is rendered,
- o Then the styles are correctly applied to the icon.

3. Accessibility:

- o **Given** an ARIA label is provided via props,
- When the Icon component is rendered,
- o **Then** the ARIA attributes are correctly added to the SVG.

4. Error Handling:

- o Given an invalid or missing iconName is provided,
- o When the Icon component is rendered,
- o Then a default icon or fallback behavior is implemented with a console warning.

5. Storybook Documentation:

- o Given the Storybook is accessed,
- o When the Icon component is previewed,
- Then examples of different icons, sizes, and colors are available for users.

Technical Tasks:

- 1. Create the Icon component with a prop (iconName) for dynamic SVG selection.
- 2. Implement prop validation to ensure valid inputs for iconName, size, and color.
- 3. Add utility functions to map iconName to SVG imports or references.
- 4. Write unit tests for:
 - Valid icon rendering.
 - o Default behavior for invalid or missing iconName.
 - Style and accessibility validations.
- 5. Add stories in Storybook showcasing:
 - o Examples of multiple icons.
 - o Prop usage for size, color, and ARIA labels.
 - o Edge cases such as invalid iconName.

Definition of Done:

- All acceptance criteria are met.
- Code is reviewed and approved.
- Unit tests achieve at least 90% coverage.
- Component passes accessibility checks (e.g., with tools like axe-core).
- Stories are added to Storybook with clear examples and documentation.
- The component is merged to the main branch with no conflicts and is ready for release.

UI Components (KEKO24-35)

IKEKO24-431 Avatar Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25

[1121102 10] 11 10 11 10 12 10 10			
Status:	Done		
Project:	<u>Keko 2024</u>		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Parent:	<u>UI Components</u>		

Type:	Task	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Abdul Sami
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Issue links:	Blocks				
	blocks	plocks <u>KEKO24-63</u> profile management page		To Do	
	blocks	<u>KEKO24-67</u>	Home page	Done	
Sprint:	KEKO24 Sprint 1				
Rank:	0 i0009v	•			

Description

User Story: Develop a Reusable Avatar Component for Website

As a frontend developer,

I want to use a reusable avatar component with configurable inputs, *so that* I can easily display different avatars without redundant code.

Acceptance Criteria:

1. **Default Avatar Rendering:**

- o Given the developer does not provide an avatar source,
- o When the avatar component is rendered,
- Then it should display a default avatar image.

2. Custom Avatar Rendering:

- o Given the developer provides a custom image source URL,
- o When the avatar component is rendered,
- o *Then* it should display the provided image correctly.

3. Avatar Size Variants:

- o Given the developer specifies a size property (e.g., small, medium, large),
- When the avatar component is rendered,
- o *Then* it should adjust its size accordingly.

4. Placeholder Support:

- o Given the provided avatar source is invalid or unavailable,
- When the component attempts to render the avatar,
- o *Then* it should fallback to a placeholder image.

5. Accessibility Compliance:

- o Given the component is used in the application,
- When it is accessed by screen readers,
- Then it should provide meaningful alt text based on the input.

6. Customizable Styles:

- o Given the developer provides additional class names or inline styles,
- o When the avatar component is rendered,
- o *Then* it should apply the provided styles correctly.

7. Click Event Handling:

- o Given the developer attaches an event handler to the avatar component,
- When the user clicks on the avatar,
- o *Then* the assigned function should be triggered.

Technical Specifications:

- The component should accept props such as src, alt, size, className, and onClick.
- Implement in React with TypeScript for type safety.
- Use default image assets stored in the /public directory.
- Support responsive design through CSS utility classes.
- Unit tests should cover edge cases like missing images or invalid inputs.

UI Components (KEKO24-35) [KEKO24-42] Rating (Stars) Component Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: UI Components

Type:	Task	Priority:	Medium		
Reporter:	Muhammad Rashid	Assignee:	Muhammad Rashid		
Resolution:	Done	Votes:	0		
Labels:	None	None			
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Issue links:	Blocks				
	blocks	<u>KEKO24-64</u>	Review page	To Do	
	blocks	<u>KEKO24-59</u>	User profile page	Done	
Sprint:	KEKO24	KEKO24 Sprint 1			
Rank:	0 i0006t:				

Description

Rating (Stars) Component

Overview

The Rating component allows developers to display a star rating system based on an input value. It is a reusable, customizable component that can be integrated into any web application.

Features

- Accepts dynamic input for rating values.
- Customizable maximum star count.
- Supports half and full stars.
- Adjustable star size and colors.
- Accessibility support with ARIA attributes.

Implementation

4. Accessibility Considerations

- Ensure aria-label conveys the rating information.
- Provide keyboard navigation if the component is interactive.

5. Additional Customization

- Add click handlers for interactive ratings.
- Provide custom SVG icons for stars.
- Allow fractional ratings with precise rendering.

Conclusion

This Rating component provides an easy-to-use and customizable way to display star ratings dynamically. Developers can integrate it efficiently by supplying the required props for different use cases.

UI Components (KEKO24-35)

KEKO24-41] Search-input component Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25

[KEKO24-41	Sear CH-III put Component Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25
Status:	Done
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None
Parent:	UI Components

Type:	Task	Priority:	Medium		
Reporter:	Desmond Deveh	Assignee:	Matti Ilvonen		
Resolution:	Done	Votes:	0		
Labels:	None	None			
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Issue links:	Blocks				
	blocks	KEKO24-64	Review page	To Do	
	blocks	<u>KEKO24-67</u>	Home page	Done	
Sprint:	KEKO24	KEKO24 Sprint 1			
Rank:	0 i0008i:i				

Description

Search Input Field Component

Title: Create a Reusable Search Input Field Component

Description:

As a developer, I want a reusable search input field component that integrates with Radix UI and React.js, so that it can be consistently used across the project for all search-related functionalities. Acceptance Criteria: Search Input Field Component

AC1: Accessible Search Input

Given the developer uses the search input field component,

When it is rendered,

Then it should include attributes such as aria-label for accessibility.

AC2: Support for Debounced Search

Given the search input field is used,

When the developer specifies a debounce time,

Then the component should delay the execution of the onChange handler by the given time.

AC3: Customization via Props

Given the developer uses the search input field component,

When they pass attributes like placeholder, value, and onChange,

Then the component should render a functional search input.

AC4: Radix UI Integration

Given the search input field component is used,

When it is rendered in the DOM,

Then it should utilize Radix UI primitives for styling and accessibility.

UI Components (KEKO24-35) [KEKO24-40] Country List Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: UI Components

Type:	Task	Priority:	Medium			
Reporter:	Uzair Irshad	Assignee:	Uzair Irshad			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified	Not Specified				
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Sprint:	KEKO24 Sprint 1
Rank:	0 i0006w:

Description

Task Description:

Develop a reusable **Country List Component** using the Radix UI library. This component should allow developers to integrate it into various parts of the application, where users need to select a country. It will support functionality for both **user information forms** and **search filters**. The component should leverage Radix UI's base components, ensuring modularity and accessibility.

Sub-Tasks:

1. Design and API Requirements

- Define the API interface for the Country List Component, allowing the developer to:
 - Input a list of countries.
 - Specify a pre-selected country (if applicable).
- Ensure support for single-selection dropdown behavior.

2. Implementation with Radix UI

- o Utilize Radix UI primitives (DropdownMenu, Select, etc.) to implement the country selection dropdown.
- Ensure the component is styled to align with the design system.

3. Accessibility

- Use ARIA-compliant attributes to ensure the component is fully accessible.
- Support keyboard navigation for country selection.

4. Integration Examples

- o Create examples of the component in two contexts:
 - **User Form:** Selecting a country for personal information.
 - **Search Filter:** Filtering data based on country selection.

5. **Testing**

- Write unit tests for the component to validate:
 - Correct rendering of the country list.
 - Functionality of the dropdown for selection.
 - Accessibility features.
- o Implement snapshot tests for visual consistency.
- Add integration tests to ensure the component works in both use cases.

6. **Documentation**

- o Provide a clear README or component documentation:
 - Usage instructions.
 - API details (props, events, etc.).
 - Example code snippets.

Acceptance Criteria:

AC1: Dropdown Functionality

- **Given** a list of countries is provided,
- When the user clicks on the dropdown,
- Then the dropdown should display the list of countries.

AC2: Pre-Selected Country

- **Given** a pre-selected country is specified,
- When the component is rendered,
- Then the dropdown should display the pre-selected country by default.

AC3: Country Selection

- **Given** the dropdown is open,
- When the user selects a country,
- **Then** the component should emit the selected country value.

AC4: Accessibility Compliance

- **Given** the component is used in a form,
- When accessed via a screen reader or keyboard,
- **Then** it should be fully navigable and provide ARIA-compliant feedback.

AC5: Reusability

- **Given** different use cases (forms, filters),
- When the component is used,

	ould support flexib	 	

UI Components (KEKO24-35) [KEKO24-39] Logo Component Created: 20/Jan/25 Updated: 07/May/25 Resolved: 24/Feb/25 Status: Done Project: Keko 2024 Components: None Affects versions: None

Fix versions: None
Parent: UI Components

Type:	Task	Priority:	Medium			
Reporter:	Rizwan Anwar	Assignee:	Rizwan Anwar			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified	Not Specified				
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Blocks	Blocks				
	blocks	<u>KEKO24-57</u>	Login page	Done		
	blocks	<u>KEKO24-58</u>	Registration page	Done		
	blocks	<u>KEKO24-61</u>	Password Reset page	Done		
Sprint:	KEKO24	Sprint 1				
Rank:	0 i0007b:					

Description

Acceptance Criteria:

- The component should accept the following **props**:
 - o logoType (string, required): Specifies which logo to render.
 - o size (string | number, optional): Controls the size of the logo.
 - o altText (string, required): Provides alt text for accessibility.
 - o className (string, optional): Allows additional styling.
 - o onclick (function, optional): Adds a click handler.
- If logoType is not provided or is invalid, a **default logo** should be displayed.
- The component should be **responsive** and **accessible** with proper altText.
- It should be **reusable** across different sections of the application.

Reusability:

- The component should be designed to be **reusable across different sections and pages**.
- It must accept the following **props** to customize its behavior:
 - o altText (string, required): Provides alt text for accessibility and SEO.

3. Responsive Design:

- The component must be **responsive**, adapting its size and alignment for:
 - o **Mobile**: Optimized display with proper scaling.
 - o **Tablet**: Well-aligned and proportionate rendering.
 - o **Desktop**: Crisp and appropriately scaled appearance.

4. Accessibility:

- All logos must include **alt text** provided through the altText prop for screen readers.
- Keyboard accessibility should be ensured for interactive logos when onClick is provided.
- Fallback logos should also be accessible and labeled correctly.

5. Documentation:

- The Logo component must be documented with:
 - o Usage examples demonstrating different logoType values.
 - o Configuration of size and altText properties.
 - o Customization with className.
 - o Interactive example showcasing onclick usage.

6. Testing and Validation:

- The component should be tested for:
 - o Cross-browser compatibility.
 - **Responsiveness** on mobile, tablet, and desktop.
 - o Accessibility checks for screen readers and keyboard interactions.
 - o Unit tests for prop handling and rendering logic.

UI Components (KEKO24-35)

[KEKO24-38] Input field component Created: 20/Jan/25 Updated: 24/Feb/25 Resolved: 24/Feb/25

[IXEIXO24-30]	Tiput_Iteru_Component Created: 20/Jan/25 Opdated: 24/Feb/25 Resolved: 24/Feb/25
Status:	Done
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None
Parent:	<u>UI Components</u>

Type:	Task	Priority:	Medium			
Reporter:	Desmond Deveh	Assignee:	Matti Ilvonen			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified					
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Blocks				
	blocks	KEKO24-64	Review page	To Do	
	blocks	<u>KEKO24-57</u>	Login page	Done	
	blocks	<u>KEKO24-58</u>	Registration page	Done	
	blocks	<u>KEKO24-61</u>	Password Reset page	Done	
Sprint:	KEKO24 Sprint 1				
Rank:	0 i0008h:				

Description

Input Field Component

Title: Create a Reusable Input Field Component

Description:

As a developer, I want a reusable input field component that integrates with Radix UI and React.js, so that it can be consistently used across the project for any input-related requirements.

Acceptance Criteria: Input Field Component

AC1: Support for Multiple Input Types

Given the developer uses the input field component,

When they specify the input type (e.g., text, email, password),

Then the component should render the appropriate input field.

AC2: Integration with Radix UI Themes

Given the input field component is used,

When it is rendered in the DOM,

Then it should utilize Radix UI primitives to ensure accessibility and consistency.

AC3: Customization via Props

Given the developer uses the component,

When they pass attributes such as placeholder, value, onChange, and required,

Then the component should apply these properties to the input field.

AC4: Styling Flexibility

Given the input field is rendered,

When custom classes are passed through the className prop,

Then the component should allow overriding or extending default styles.

UI Components (KEKO24-35) | KEKO24-37 | Button component | Created: 20/Jan/25 | Updated: 24/Feb/25 | Resolved: 24/Feb/25 | | Status: | Done | | Project: | Keko 2024 | | Components: | None | | Affects versions: | None | | Fix versions: | None | | Parent: | UI Components |

Type:	Task	Priority:	Medium			
Reporter:	Matti Ilvonen	Assignee:	Matti Ilvonen			
Resolution:	Done	Votes:	0			
Labels:	None	None				
Remaining Estimate:	Not Specified	Not Specified				
Time Spent:	Not Specified					
Original estimate:	Not Specified					

Issue links:	Blocks	Blocks					
	blocks	KEKO24-64	Review page	To Do			
	blocks	KEKO24-57	Login page	Done			
	blocks	KEKO24-58	Registration page	Done			
	blocks	<u>KEKO24-59</u>	User profile page	Done			
	blocks	<u>KEKO24-61</u>	Password Reset page	Done			
	blocks	<u>KEKO24-67</u>	Home page	Done			
Sprint:	KEKO2	KEKO24 Sprint 1					
Rank:	0 i0008i:						

Description

Create a reusable button and extensible button component within the comonent library. The Button component is a control element. It allows users to perform actions and make selections with a single click. It has different states: enabled, hover, pressed, disable. It should support customization options such as name, size, colour, font.

```
import Button from './Button';

/* Example how Button component would be used in code */
<Button
  label="Home"
  onClick={() => console.log('Navigating to Home')}
    ariaLabel="Go to Home page"
  size="medium"
  color="primary"
/>
```

Component Features

1. Base Functionality:

- o Implements the <button> HTML element as the base.
- o Supports multiple states: enabled, hover, pressed, disabled.
- Triggers actions on click events.

2. Customization Options:

- o **Props**:
 - name: Button identifier for screen readers.
 - size: Predefined size options (small, medium, large).
 - color: Variants such as primary, secondary, tertiary, danger, etc.
 - disabled: Boolean to disable the button.

Output Opposition Op

- Leverage Tailwind CSS utility classes for styles.
- Merge custom classes with default styles.

o ARIA Support:

- aria-label for accessible names.
- aria-disabled for disabled state.
- o Additional props to pass custom data attributes (e.g., data-*).

3. Radix-UI Integration:

 Utilize Radix-UI primitives to ensure consistent keyboard navigation, focus states, and accessibility.

4. **Default Styles**:

- o Tailwind CSS to define default and variant-based styles for:
 - Background color.
 - Font size and weight.
 - Padding and border radius.
 - Hover and active states.

5. Extensibility:

- o Allow developers to pass additional props for extending functionality.
- o Accept className to override or extend default styling.

States and Interactions

1. **Enabled**: Normal appearance based on size and color props. Example Tailwind styles:

bg-blue-500 text-white hover:bg-blue-600 focus:ring-2 focus:ring-blue-400

1. **Hover**: Button changes style when hovered over. Example Tailwind styles:

hover:bg-opacity-80

1. **Pressed**: Button appears "pressed" during user interaction. Example Tailwind styles:

active:scale-95 active:bg-blue-700

1. **Disabled**: Button is non-interactive and grayed out. Example Tailwind styles:

Unit Testing Acceptance Criteria

1. Rendering Tests:

- o Button renders correctly with all required props.
- o Supports additional custom classes without affecting default behavior.

2. State Tests:

o Button applies correct styles for enabled, hover, pressed, and disabled states.

3. **Prop Tests**:

- o Button accepts and correctly applies name, size, color, font, and other props.
- o Handles aria-label and aria-disabled attributes for accessibility.

4. Interaction Tests:

- o Simulate user clicks and verify the correct callback is triggered.
- o Ensure disabled buttons do not trigger actions.

Storybook Stories

1. **Default Button**:

o A basic example of the button with default settings.

2. Variants:

- o Showcases different color options (e.g., primary, secondary, danger).
- o Demonstrates size variations (small, medium, large).

3. States:

o Separate stories for hover, pressed, and disabled states.

4. Custom Class Example:

o Demonstrates overriding styles using the className prop.

5. Accessible Button:

o Example with aria-label and usage in a form or modal context.

Defenition of done:

- Component renders correctly and handles props as specified.
- Visual and functional tests written and passing.
- Documentation and Storybook entries are complete.
- Reviewed, merged, and deployed without critical issues.

[KEKO24-36] A	[KEKO24-36] Avatar Created: 20/Jan/25 Updated: 17/Feb/25				
Status:	To Do				
Project:	<u>Keko 2024</u>				
Components:	None				
Affects versions:	None				
Fix versions:	None				

Type:	Story	Priority:	Medium		
Reporter:	Abdul Sami	Assignee:	Unassigned		
Resolution:	Unresolved	Votes:	0		
Labels:	None				
Remaining Estimate:	Not Specified				
Time Spent:	Not Specified				
Original estimate:	Not Specified				

Issue links:	Blocks	Blocks				
	blocks	<u>KEKO24-59</u>	User profile page	Done		
Sprint:						
Rank:	0 i0006n:					

[KEKO24-35] UI Components Created: 20/Jan/25 Updated: 25/May/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Epic	Priority:	Medium
Reporter:	Matti Ilvonen	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

UI Components are the basic components like:

- Logo <u>Rizwan Anwar</u>
- Avatar Abdul Sami
- Icon Matti Ilvonen
- Button Kseniia Kochetova
- profile picture Abdul Sami
- Rating (stars) Muhammad Rashid
- Badge Anastasiia
- Input Filed <u>Desmond Deveh</u>
- Search <u>Desmond Deveh</u>
- Drowdown Maryam Agaiby
- Modal Arkar Myint Myat
- Textarea Input Field Anastasiia
- Country List <u>Uzair Irshad</u>
- Date Picker Uzair Irshad

Everyone's task is to create Jira tasks for each component that is assigned to them. Use the format we used in user stories. Use acceptance criteria. Link to the Figma screen where the component is used. If component is used in many screens, you don't have to link all screens.

[KEKO24-34] User Story for Rating Filter Created: 17/Jan/25 Updated: 17/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Uzair Irshad	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i00067:

User Story: As a user, I want to filter search results by freelancer rating, so that I can find services from freelancers with proven quality and performance.

Acceptance Criteria:

1. Filter Setup

- o Given I am on the search results page,
- When I access the rating filter,
- Then I should be able to select a minimum rating threshold (e.g., 3 stars, 4 stars).

2. Custom and Predefined Ranges

- o Given I want to filter by rating,
- o When I select a predefined range or manually input a custom threshold,
- o **Then** the results should update to show freelancers with an average rating that meets or exceeds the specified threshold.

3. Visibility and Usability

- o Given I am using the rating filter,
- o When I interact with it,
- o **Then** it should be clearly visible, easy to use, and provide clear feedback on selected ranges.

4. Accurate Reflections

- o Given I filter by rating,
- o When I view the freelancer profiles in the results,

0	Then their displayed ratings should accurately reflect the platform's calculation based on past user reviews.

[KEKO24-33] Password update form Created: 13/Jan/25 Updated: 20/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Kseniia Kochetova	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0005z:

Description

As a

User,

I want

to update my password from recovery link,

So that

I can regain access to my account.

Acceptance Criteria:

Update Password:

- 1. **Given** I click the reset link in my email,
- 2. When I set a new password and confirm it,
- 3. **Then** my password should be updated, and I should be redirected to the login page.

[KEKO24-32] Location Filter Created: 28/Dec/24 Updated: 05/May/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Rizwan Anwar	Assignee:	Uzair Irshad
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0005r:

As a user,

I want to filter search results by location (e.g., country, city), so that I can find freelancers who are in my region or offer services in my time zone.

Acceptance Criteria:

- The location filter allows users to select their preferred location or region (e.g., United States, Europe, India, etc.).
- Depending on the platform's coverage, the filter can be based on country, city, or region.
- Users can select a single location or multiple locations to refine search results.
- The location filter is clearly labeled and easy to use.
- The filter ensures that freelancers' location information is accurately reflected in search results.

[KEKO24-31] Gig Category Filter Created: 28/Dec/24 Updated: 28/Dec/24	
Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Rizwan Anwar	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0005j:

As a user,

I want to filter search results by gig category (e.g., Graphic Design, Writing, Programming), so that I can find services related to a specific skill or area of expertise.

Acceptance Criteria:

- Gig categories are displayed as a list or dropdown (e.g., Web Development, Marketing, Graphic Design, etc.).
- The categories should be based on pre-defined classifications of the available services.
- Users can select one or more categories to narrow down search results.
- The category filter is visible and accessible within the search interface.

[KEKO24-30] Seller Level Filter Created: 28/Dec/24 Updated: 05/May/25	
Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Rizwan Anwar	Assignee:	<u>Anastasiia</u>
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0005b:
Participants:	Uzair Irshad

As a user,

I want to filter search results by seller level (e.g., Beginner, Intermediate, Expert), so that I can choose a freelancer based on their experience and reputation.

Acceptance Criteria:

- The seller level filter includes options such as "New Seller," "Level 1," "Level 2," and "Top Rated."
- The filter is clearly labeled and easily accessible in the search interface.
- Users can select one or more seller levels to refine search results.
- Seller levels are visible in the seller profile and are accurately reflected in search results.

[KEKO24-29] Price Range Filter Created: 28/Dec/24 Updated: 05/May/25	
Status:	To Do
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Rizwan Anwar	Assignee:	<u>Anastasiia</u>
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i00053:
Participants:	Uzair Irshad

User Story: Filter Search Results by Price Range

As a user,

I want to filter search results by price range, so that I can find services that fit within my budget.

Acceptance Criteria

1. Filter by Price Range

- o Given the user is on the search results page,
- o When they set a minimum and maximum price in the price filter,
- Then the search results are updated to show only services within the specified price range.

2. Predefined Price Ranges

- o **Given** the user is on the search results page,
- o When they select a predefined price range (e.g., \$10–\$50 or \$50–\$100),
- o **Then** the search results are updated to reflect the selected range.

3. Custom Price Values

- o **Given** the user is on the search results page,
- o When they enter custom minimum and maximum price values in the filter,
- o **Then** the search results are updated to reflect the entered price range.

4. Filter Display

o Given the user is on the search results page,

- o When the page is loaded,
- o **Then** the price filter is displayed clearly, with a user-friendly interface that allows easy interaction.

5. Application to Total Cost

- o **Given** the user applies a price filter,
- o When services include additional charges or fees,
- Then the total cost, including any extras, is used to filter the results.

This story aligns with best practices for user stories and includes testable acceptance criteria in the Given/When/Then format [6†source] [7†source].

[KEKO24-28] User: Add CV or Other Files to Profile to Highlight	
Professionalism	Created: 17/Dec/24 Updated: 17/Dec/24
Status:	To Do
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0004v:

User Story:

As a freelancer, I want to upload my CV and other files (e.g., cover letter) to my profile, so that I can showcase my professionalism and improve my chances of attracting clients.

Acceptance Criteria:

1. Navigate to CV and File Upload Section:

- o **Given** I am on my published profile page,
- When I navigate to the "Edit Profile" section and select "Add CV & Other Files."
- Then I should be redirected to the media upload section.

2. Upload CV or Other Files:

- o **Given** I am on the media upload section,
- When I upload valid files (e.g., PDF, DOCX) for my CV or other documents,
- Then the system should accept the files and display them as successfully uploaded.

3. Preview Uploaded Files:

- o **Given** I have uploaded my CV or other files,
- When the upload is successful,
- Then I should see a preview or confirmation of the uploaded file names.

4. Handle Invalid File Uploads:

Given I attempt to upload a file that is in an unsupported format or exceeds the file size limit,

- o When I click on "Upload,"
- o **Then** I should see an error message explaining the issue, and the file should not be uploaded.

5. Save and Display Uploaded Files:

- o Given I have successfully uploaded my CV or other files,
- o When I click the "Save" button,
- Then the files should be saved and displayed in the "CV & Other Files" section on my profile.

6. Edit or Remove Files:

- o **Given** I have already uploaded files to my profile,
- When I navigate to the "Edit Profile" section and choose to update or delete the files,
- o **And** I click the "Save" button,
- o **Then** the changes should be reflected immediately on my profile.

[KEKO24-27] <u>User: Add Certificates to Profile to Showcase Professionalism</u> Created: 17/Dec/24 Updated: 17/Dec/24	
Status:	To Do
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0004n:

User Story:

As a freelancer, I want to update my profile page with my certificates, so that I can showcase my professionalism and build trust with potential clients.

Acceptance Criteria:

1. Navigate to Certificate Upload Section:

- o Given I am on my published profile page,
- o When I navigate to the "Edit Profile" section and select "Add Certificates,"
- Then I should be redirected to the media upload section for certificates.

2. Upload Certificate Files:

- o Given I am on the media upload section for certificates,
- o When I upload a valid certificate file (e.g., PDF, PNG, JPEG),
- **Then** the system should accept the file and display a preview of the uploaded certificate.

3. Handle Invalid Certificate Files:

- o **Given** I am on the media upload section for certificates,
- When I attempt to upload a file that is not a supported format or exceeds the allowed file size,
- Then I should see an error message indicating the issue, and the file should not be uploaded.

4. Save and Display Certificates:

- o **Given** I have successfully uploaded a certificate file,
- o When I click the "Save" button,

• **Then** the uploaded certificate should be saved and displayed under the "Certificates" section on my profile.

5. Edit or Remove Certificates:

- o Given I have certificates already listed in my profile,
- When I navigate to the "Edit Profile" section and choose to update or delete a certificate,
- o And I click "Save,"
- o Then the changes should reflect immediately on my profile

[KEKO24-26] <u>User(Freelancer): Add Educational Background to Profile</u> Created: 17/Dec/24 Updated: 19/Jan/25		
Status:	To Do	
Project:	Keko 2024	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0004f:

User Story:

As a freelancer, I want to update my profile page with my educational background, so that I can showcase my qualifications and professionalism to clients.

Acceptance Criteria:

1. Add Educational Background:

- o **Given** I am on my published profile page,
- When I navigate to the "Edit Profile" section and select the "Add Education" option,
- And I enter my educational background details (e.g., degree, institution name, and year of completion),
- o And I click the "Save" button,
- Then my educational background should be visible on my profile under the "Education" section.

2. Edit Existing Educational Information:

- o **Given** I already have education details listed on my profile,
- When I navigate to the "Edit Profile" section and update or remove my educational background information,
- o And I click the "Save" button,
- Then the updated educational details should be reflected on my profile.

3. Validate Required Fields:

o Given I am adding or editing my educational background,

- When I leave any required field (e.g., degree or institution name) blank and click "Save,"
- Then I should see an error message prompting me to fill in the missing fields.

4. Display Education on Profile:

- o Given I have successfully added my educational background,
- o When I view my public profile,
- o **Then** the "Education" section should display the entered details, including the degree, institution name, and year of completion.

[KEKO24-25] User: Update Profile with Skills to Showcase Expertise Created: 17/Dec/24 Updated: 19/Jan/25	
Status:	To Do
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i00047:

User Story:

As a freelancer, I want to update my profile with my skills, so that I can showcase my expertise and make my profile attractive to clients.

Acceptance Criteria:

1. Add Skills to Profile:

- o **Given** I am on my published profile page,
- When I navigate to the "Edit/Update Profile" section,
- o And I add my skills to the "Skills" section,
- o And I click the "Save" button,
- Then the newly added skills should be visible on my profile.

2. Edit Existing Skills:

- o **Given** I already have skills listed on my profile,
- o When I navigate to the "Edit/Update Profile" section,
- o And I modify or remove skills in the "Skills" section,
- o And I click the "Save" button,
- o Then the updated skills should be reflected on my profile.

3. Error Handling for Missing Skill Input:

- o **Given** I attempt to save my profile without adding any skills to the "Skills" section
- o When I click the "Save" button,
- o **Then** I should see an error message prompting me to add at least one skill.

4. Skills Display on Profile:

- Given I have successfully saved my skills,
 When I view my public profile,
 Then my skills should be displayed clearly under the "Skills" section.

[KEKO24-24] <u>User/Client: Control Visibility of Service or Task Listings</u> Created: 16/Dec/24 Updated: 16/Dec/24		
Status:	To Do	
Project:	Keko 2024	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium	
Reporter:	Desmond Deveh	Assignee:	Unassigned	
Resolution:	Unresolved	Votes:	0	
Labels:	None	None		
Remaining Estimate:	Not Specified			
Time Spent:	Not Specified			
Original estimate:	Not Specified			

Sprint:	
Rank:	0 i0003z:

ser Story:

As a user, I want to set the visibility of my service or task listings to public or private, so that I can control who can view and request my services.

Acceptance Criteria:

1. Set Visibility During Creation:

- o Given I am on the "Create Job/Gig/Project" form,
- When I select a visibility option (public or private) while filling out the form,
- o And I click "Save,"
- o **Then** my service listing should only be visible to users based on my selected option (public or private).

2. Update Visibility for Existing Listings:

- o **Given** I have an existing service listing,
- o When I edit the listing and change the visibility option (public or private),
- o And I click "Save,"
- Then the updated visibility settings should immediately take effect for that listing.

3. Public Visibility Behavior:

- o **Given** I have set my listing to "Public,"
- o When other users browse the platform,
- o **Then** my listing should be visible and accessible to all users.

4. Private Visibility Behavior:

o **Given** I have set my listing to "Private,"

- o When other users browse the platform,
- Then my listing should not appear in public searches and should only be accessible to users I explicitly share it with (if applicable).

5. **Default Visibility Option:**

- o Given I am creating a new service listing,
- o When I do not actively select a visibility option,
- o **Then** the system should set a default visibility (e.g., public or private as defined by platform settings) and notify me of this default setting.

[KEKO24-23] <u>User/client: Add Multiple Jobs or Tasks to Listings</u> Created: 16/Dec/24 Updated: 19/Jan/25		
Status:	To Do	
Project:	Keko 2024	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium	
Reporter:	Desmond Deveh	Assignee:	Unassigned	
Resolution:	Unresolved	Votes:	0	
Labels:	None	None		
Remaining Estimate:	Not Specified			
Time Spent:	Not Specified			
Original estimate:	Not Specified			

Sprint:	
Rank:	0 i0003r:
Flagged:	Impediment

User Story:

As a client, I want to add multiple jobs or tasks to my listing in one session, so that I can efficiently attract freelancers to offer their services.

Acceptance Criteria:

1. Add Initial Job/Task Card:

- o **Given** I am on the "Create Job/Task" form,
- o When I enter details such as title, description, pricing, and delivery time,
- o **And** I click the "Add More" button,
- **Then** I should be redirected to a new "Create Job/Task" form to add another card.

2. Add Additional Job/Task Cards:

- Given I have added one or more tasks and remain on the "Create Job/Task" flow,
- o When I repeat the process of entering details and clicking "Add More,"
- o **Then** each job/task should be queued for publishing without leaving the flow.

3. Publish All Tasks:

- o **Given** I have added one or more jobs/tasks,
- o When I click "Create" to finalize,
- o Then all tasks should be published together and displayed in my listing.

4. Error Handling for Missing Details:

- o Given I attempt to add a job/task with incomplete details,
- o When I click "Add More" or "Create,"
- Then I should see an error message, and the task should not be added or published until the issue is resolved.

Comments

Comment by Desmond Deveh [19/Jan/25]

Flag added

Has the same functionality as 22

[KEKO24-22] User / client: Add Jobs or Projects Created: 16/Dec/24 Updated: 19/Jan/25		
Status:	To Do	
Project:	Keko 2024	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0003j:

User Story:

As a client, I want to add a job, gig, or project to my listing, so that I can upsell extras to clients or freelancers.

Acceptance Criteria:

1. Create a Job/Project Card:

- o **Given** I am on the "Create Job/Project" form,
- When I enter all required details, including title, description, pricing, and estimated delivery time,
- o **And** I click the "Create" button,
- Then the job/project should be successfully created and displayed in my listing as a published card.

2. Error Handling for Missing Fields:

- o Given I am on the "Create Job/Project" form,
- o When I try to create a job/project without filling all required fields,
- Then I should see an error message specifying which fields need to be completed, and the job/project should not be published.

3. Edit Job/Project Card:

- o Given I have an existing job/project card in my listing,
- When I navigate to the card and update the fields (e.g., title, description, pricing),
- o And I click "Save,"

o **Then** the changes should be reflected immediately, and the card should remain published.

4. View Published Job/Project:

- o Given I have successfully created a job/project card,
- o When I navigate to my listing,
- o **Then** I should see the newly created job/project displayed with all entered details.

[KEKO24-21] Enhanced Search Filters for Freelance Services Created: 16/Dec/24 Updated: 16/Dec/24		
Status:	To Do	
Project:	Keko 2024	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Uzair Irshad	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0003b:

1. Price Range Filter

User Story:

As a user, I want to filter search results by price range, so that I can find services that fit within my budget.

Acceptance Criteria:

1. Filter Setup

- o Given I am on the search results page,
- o When I access the price filter,
- o **Then** I should be able to set a minimum and maximum price.

2. Custom and Predefined Ranges

- o **Given** I want to filter by price,
- When I enter custom values or select predefined price ranges (e.g., \$10-\$50, \$50-\$100),
- o **Then** the results should update to show only services within the specified range.

3. Visibility and Usability

- o Given I am using the price filter,
- o When I interact with it,
- o **Then** it should be displayed clearly and function seamlessly.

4. Total Cost Inclusion

- o **Given** I filter by price,
- o When a service includes additional charges or fees,
- o **Then** the price filter should account for the total cost of the service.

2. Seller Level Filter

User Story:

As a user, I want to filter search results by seller level (e.g., Beginner, Intermediate, Expert), so that I can choose a freelancer based on their experience and reputation.

Acceptance Criteria:

1. Filter Options

- o **Given** I access the seller level filter,
- o When I view the available options,
- Then I should see levels such as "New Seller," "Level 1," "Level 2," and "Top Rated."

2. Multiple Selections

- o Given I want to refine search results,
- When I select one or more seller levels.
- o **Then** only results matching those levels should be displayed.

3. Visibility and Accessibility

- o Given I am on the search interface,
- o When I look for the seller level filter,
- o **Then** it should be clearly labeled and easy to find.

4. Accurate Reflections

- o **Given** I filter by seller level,
- When I view the seller profiles in the results,
- o **Then** their levels should be accurately reflected based on the platform's criteria.

3. Gig Category Filter

User Story:

As a user, I want to filter search results by gig category (e.g., Graphic Design, Writing, Programming), so that I can find services related to a specific skill or area of expertise.

Acceptance Criteria:

1. **Display Options**

- o Given I access the category filter,
- o **When** I view the list or dropdown,
- Then I should see predefined classifications like "Web Development,"
 "Marketing," and "Graphic Design."

2. Multiple Selections

- o **Given** I want to refine my search,
- o When I select one or more categories,
- o **Then** the results should display services only in those categories.

3. Visibility and Accessibility

- o **Given** I am on the search interface,
- o When I look for the category filter,
- o **Then** it should be visible and easy to use.

4. Accurate Categorization

- o **Given** I filter by category,
- o When I view the results,
- o **Then** the displayed services should match the selected categories.

4. Location Filter

User Story:

As a user, I want to filter search results by location (e.g., country, city), so that I can find freelancers who are in my region or offer services in my time zone.

Acceptance Criteria:

1. Filter Setup

- o Given I access the location filter,
- o **When** I view the options,
- o **Then** I should be able to select a specific country, city, or region (e.g., United States, Europe, India).

2. Single or Multiple Locations

- o Given I want to refine my search,
- When I select one or more locations,
- o **Then** the results should display freelancers from the selected areas.

3. Visibility and Usability

- o **Given** I am using the location filter,
- o When I interact with it,
- o **Then** it should be clearly labeled and easy to use.

4. Accurate Reflections

- o Given I filter by location,
- o **When** I view the results,
- o **Then** the freelancers' location information should be accurate and align with the selected filter.

[KEKO24-20] Account Registration Created: 16/Dec/24 Updated: 20/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 hzzzzi:

As a

New User,

I want

to register my email and password,

So that

I can create an account.

Acceptance Criteria:

1. Valid Registration:

- o Given I am on the registration page,
- o When I provide a valid email, password, and confirm the password,
- Then I should proceed to the next step of the registration process.

2. Weak Password Validation:

- o **Given** I am on the registration page,
- o When I enter a weak password,
- Then I should see an error message prompting me valid parameters I need for the password (length, complexity, etc.).

3. Mismatched Passwords:

- o Given I enter mismatched passwords,
- o **When** I submit the form,
- Then I should see an error message prompting me to re-enter matching passwords.

4. Empty Fields Validation:

- o **Given** any required field is left empty,
- o When I click the Register button,
- Then I should see an error message prompting me to fill it.

5. Duplicate Email Handling:

- o **Given** I provide an already-registered email,
- o **When** I submit the form,
- o Then I should see an error message saying, "Email already registered."

[KEKO24-19] <u>User: Add Portfolio Samples to Service Listings</u> Created: 16/Dec/24 Updated: 19/Jan/25	
Status:	To Do
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0003h:

User Story:

As a user, I want to add portfolio samples to my service listing, so that I can showcase my expertise to potential clients.

Acceptance Criteria:

1. Upload Portfolio Media:

- o **Given** I am on the "Add Media" form for my service listing,
- When I click on "Upload" or drag and drop media files,
- And the media file is of an acceptable type (e.g., image, video, document),
- Then the media should be successfully uploaded and displayed in the media preview section.

2. Preview Uploaded Media:

- o **Given** I have uploaded a media file,
- o When I view the preview of the uploaded file,
- o **Then** the preview should match the uploaded media, ensuring it is displayed correctly.

3. Publish Service with Media:

- o Given I have uploaded one or more portfolio media files,
- o When I click the "Publish" button on my service listing,
- Then the media files should be published alongside my service listing, viewable to potential clients.

4. Error Handling for Invalid Media Files:

o Given I try to upload a media file of an unsupported type or size,

- When I click "Upload" or drag and drop the file,
 Then I should see an error message specifying the issue, and the file should not be uploaded.

[KEKO24-18] View Freelancer's Reviews Created: 16/Dec/24 Updated: 18/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Muhammad Rashid	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i00006:

As a client,

I want to see reviews left by other clients on a freelancer's profile, so that I can decide whether to hire them.

Acceptance Criteria:

1. View Reviews Section:

Given I am on the freelancer's profile page,

when I scroll down to the reviews section,

then I should see a list of reviews from other clients, each displaying:

- o The rating (e.g., stars or numerical scale),
- o Review text,
- o Date of completion of the project,
- o And the date the review was submitted.

2. Sort Reviews:

Given I am in the reviews section,

when I choose a sorting option,

then the reviews should be sorted according to the selected parameter.

Sorting Options:

- o By date (from newest to oldest),
- o By date (from oldest to newest),
- o By rating (from highest to lowest),
- By rating (from lowest to highest).

3. Search Reviews by Keywords or Categories:

Given I am in the reviews section,

when I enter a keyword in the search bar (e.g., "design," "website," "marketing"), **then** the system displays only reviews containing:

- o The entered keyword in the review text,
- o The client's name,
- o Or projects or categories mentioned in the review.

4. Filter Reviews by Parameters:

Given I am in the reviews section,

when I select filter parameters,

then the system displays only reviews that match the selected criteria.

[KEKO24-17] Password Recovery Process Created: 16/Dec/24 Updated: 20/Jan/25	
Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 hzzzza:

As a

User,

I want

to reset my password if I forget it,

So that

I can regain access to my account.

Acceptance Criteria:

1. Navigation to Reset Page:

- o **Given** I am on the login page,
- o When I click "Forgot Password,"
- Then I should be redirected to a reset password page.

2. Submit Registered Email:

- o Given I provide a registered email,
- o **When** I submit the reset form,
- Then I should receive an email with a password reset link.

3. Submit Unregistered Email:

- o Given I provide an unregistered email,
- o When I submit the reset form,
- o **Then** I should see an error message saying, "If email has been registered, we will send you a message soon."

4. Update Password:

o **Given** I click the reset link in my email,

- When I set a new password and confirm it,
- **Then** my password should be updated, and I should be redirected to the login page.

5. Expired or Invalid Reset Link:

- o Given the reset link is expired or invalid,
- When I try to use it,
- Then I should see an error message saying, "Reset link expired. Please request a new one."

[KEKO24-16] L	Enable Profile Customization for Users Created: 16/Dec/24 Updated: 16/Dec/24
Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Uzair Irshad	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0002f:

User Story:

As a user, I want to customize my profile, so that I can showcase my skills and attract more opportunities.

Acceptance Criteria:

1. Profile Picture and Bio Editing

- o **Given** I am logged in,
- o **When** I edit my profile,
- Then I should be able to upload a profile picture and write a bio.

2. Skill Tags and Portfolio

- o **Given** I am a seller,
- o When I add skill tags and upload portfolio samples,
- Then these should be displayed publicly on my profile.

3. Availability Status Update

- o Given I want to update my availability,
- o When I set my profile status to "Available for work" or "Away,"
- o **Then** other users should see the updated status.

[KEKO24-15] A	Admin Ability to View All Categories Created: 16/Dec/24 Updated: 18/Jan/25
Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i00004:

User Story:

As an admin, I want to view all existing categories so that I can easily understand and manage my organizational structure.

Acceptance Criteria:

1. Display List of Categories

- o Given I am on the category management page,
- o When the page loads,
- o **Then** I should see a list of all categories with their names.

2. No Categories Available

- o Given there are no categories created,
- o When I access the category management page,
- o Then I should see a message indicating no categories are available.

3. Scrollable List for Many Categories

- o Given there are many categories,
- When I view the list,
- o **Then** the categories should be scrollable for better usability.

[KEKO24-14] <u>/</u>	Admin Ability to Delete Categories Created: 16/Dec/24 Updated: 18/Jan/25
Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i00002:

User Story:

As an admin, I want to delete a category so that I can remove unnecessary categories.

Acceptance Criteria:

1. Confirmation Prompt for Deletion

- o Given I am on the category management page,
- o When I select a category and click the "Delete" button,
- Then I should see a confirmation prompt asking if I am sure about deleting the category.

2. Successful Deletion

- o Given I confirm the deletion in the prompt,
- o When I proceed,
- o **Then** the category should be removed from the category list.

User: Ability to Create and Update Service Listings (KEKO24-8) [KEKO24-13] Ability to Create and Update Service Listings Created: 16/Dec/24 Updated: 16/Dec/24 Status: To Do Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: User: Ability to Create and Update Service Listings

Type:	Subtask	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0001r:

Description

User Story:

As an account owner, I want to create and update a service listing, so that I can showcase my skills to attract clients or find freelancers with the right skills.

Acceptance Criteria:

1. Create a New Service Listing:

- o Given I am logged into my account,
- When I navigate to the "Create Service Listing" page and fill in all required fields (e.g., title, description, category, pricing, delivery time), and click "Save,"
- **Then** the service listing should be created and appear in my profile as a published service.

2. Modify an Existing Service Listing:

- o **Given** I have an existing service listing,
- When I navigate to the "Edit Service" page, make changes to the fields, and click "Save,"
- o **Then** the updated listing should reflect the changes and remain published.

3. Error Handling for Missing Required Fields:

- o Given I try to save a service listing without completing all required fields,
- o When I click "Save,"

0	Then I should see an error message specifying the missing fields, and the selisting should not be saved.

[KEKO24-12] Admin Ability to Edit Categories Created: 16/Dec/24 Updated: 18/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 hzzzzy:

User Story:

As an admin, I want to edit an existing category so that I can update its details to keep the information current.

Acceptance Criteria:

1. Access to Edit Option

- o Given I am on the category management page,
- o When I click on an existing category,
- o Then I should see an option to edit the category.

2. Successful Category Update

- o Given I modify the category name or description with valid inputs,
- o When I click "Save,"
- o **Then** the updated information should be reflected in the category list.

3. Validation for Blank or Invalid Inputs

- o Given I leave the name field blank or input invalid data,
- When I click "Save,"
- o **Then** I should see an error message specifying the required corrections.

[KEKO24-11] Ability to Edit a Review Created: 16/Dec/24 Updated: 18/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Muhammad Rashid	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0000i:

As a client,

I want to be able to edit my review,

so that I can correct mistakes or add more information if needed.

Acceptance Criteria:

1. Access Edit Review Option:

Given I have written a review and want to edit it,

when I click the "Edit" button next to my review at the bottom of the freelancer's profile page,

then an editing field appears where I can modify the text of the review, along with a "Save" button to confirm the changes.

2. Save Edited Review:

Given I have edited my review,

when I save the changes by clicking the "Save" button,

then I should see a confirmation message below the review stating, "Your review has been successfully updated."

3. Validation for Edited Review Length:

a. **Given** I am editing a review,

when I save the changes with a review between 10 and 500 characters in length, **then** the system should successfully accept the changes and display the confirmation message.

b. Given I am editing a review,

when I save the changes with a review containing less than 10 characters,

then the system should display an error message stating, "Review must be at least 10 characters long."

c. Given I am editing a review,

when I save the changes with a review containing more than 500 characters, **then** the system should display an error message stating, "Review must not exceed 500 characters."

4. Cancel Edit:

Given I am editing a review, when I click the "Cancel" button,

then the editing field should close without saving any changes, and the original review should remain unchanged.

[KEKO24-10] Admin Ability to Create Minor Categories Created: 16/Dec/24 Updated: 18/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0000s:

User Story:

As an admin, I want to create minor categories within subcategories so that I can add further levels of organization.

Acceptance Criteria:

1. Minor Category Creation Form

- o Given I am on the category management page,
- o When I select a subcategory and click "Add Minor Category,"
- o Then I should see a form to enter the minor category name and description.

2. Successful Minor Category Creation

- o Given I fill out the form with valid inputs,
- o When I click "Save,"
- o **Then** the new minor category should be created and visible under the selected subcategory in the category list.

3. Duplicate Minor Category Name Handling

- o **Given** I attempt to create a minor category with a name that already exists within the same subcategory,
- o When I click "Save,"
- Then I should see an error message indicating that the minor category name is already in use.

4. Nested Display of Minor Categories

- o Given there are minor categories created under a subcategory,
- o **When** I view the category list,



[KEKO24-9] Admin Ability to Create Subcategories Created: 16/Dec/24 Updated: 18/Jan/25		
Status:	To Do	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0000q:

User Story:

As an admin, I want to create subcategories within a main category so that I can further organize and structure my items or tasks.

Acceptance Criteria:

1. Subcategory Creation Form

- o Given I am on the category management page,
- o When I select a main category and click "Add Subcategory,"
- o **Then** I should see a form to enter the subcategory name and description.

2. Successful Subcategory Creation

- o Given I fill out the form with valid inputs,
- o When I click "Save,"
- Then the new subcategory should be created and visible under the main category in the category list.

3. Duplicate Subcategory Name Handling

- o **Given** I attempt to create a subcategory with a name that already exists within the same main category,
- o When I click "Save,"
- Then I should see an error message indicating that the subcategory name is already in use.

4. Nested Display of Subcategories

- o Given there are subcategories created under a main category,
- o When I view the category list,



[KEKO24-8] Us	ser: Ability to Create and Update Service Listings Created:
Status:	To Do
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Desmond Deveh	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	KEKO24-13	Ability to Create and Update Service	Subtask	To Do	
Sprint:					
Rank:	0 i0003f:				

to add User User Story:

As an account owner, I want to create and update a service listing, so that I can showcase my skills to attract clients or find freelancers with the right skills.

Acceptance Criteria:

1. Create a New Service Listing:

- o Given I am logged into my account,
- o **When** I navigate to the "Create Service Listing" page and fill in all required fields (e.g., title, description, category, pricing, delivery time), and click "Save,"
- Then the service listing should be created and appear in my profile as a published service.

2. Modify an Existing Service Listing:

- o **Given** I have an existing service listing,
- When I navigate to the "Edit Service" page, make changes to the fields, and click "Save,"
- o **Then** the updated listing should reflect the changes and remain published.

3. Error Handling for Missing Required Fields:

o Given I try to save a service listing without completing all required fields,

- When I click "Save,"

 Then I should see an error message specifying the missing fields, and the service listing should not be saved. Stories

[KEKO24-7] <u>S</u>	ecure User Login Created: 16/Dec/24 Updated: 05/May/25 Resolved: 05/May/25
Status:	Done
Project:	Keko 2024
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Muhammad Rashid
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 hzzzz2:

As a

User,

I want

to log in with my email and password,

So that

I can securely access my account.

Acceptance Criteria:

1. Successful Login:

- o Given I am on the login page,
- o When I enter a valid email and password,
- Then I should be logged in and redirected to my dashboard.

2. Invalid Credentials:

- o Given I enter invalid credentials,
- When I click the login button,
- Then I should see an error message prompting me "Email or password are incorrect."

3. Empty Fields Validation:

- o Given email or password fields are left empty,
- o When I click the login button,
- Then I should see an error message prompting me "Fill required fields."

4. Navigate to Registration:

o Given I am not registered,

- When I click "Registration,"
 Then I should be redirected to the registration page.

Wrapper Components (KEKO24-55) [KEKO24-6] Ability to Leave a Review Created: 16/Dec/24 Updated: 05/May/25 Status: To Do Project: Keko 2024 Components: None Affects versions: None Fix versions: None Parent: Wrapper Components

Type:	Story	Priority:	Medium
Reporter:	Muhammad Rashid	Assignee:	Abdul Sami
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	KEKO24 Sprint 3
Rank:	0 i000if:
Participants:	Abdul Sami, Desmond Deveh

Description

As a client who has hired a freelancer,

I want to have the ability to leave a review about the completed work, so that other users can learn about the quality of the freelancer's services and help them improve their reputation.

• View "Leave a Review" Button:

Given I am a client who has hired a freelancer and completed a project, **when** I visit the freelancer's profile,

then I can see a button labeled "Leave a review" to write a review.

• Access Review Form:

Given I clicked on the "Leave a Review" button,

when I navigate to the review page,

then I see:

- o Four criteria for rating the freelancer on a 5-point scale,
- o A text area for a written review,
- o And a "Submit" button.

• Submit Review for Moderation:

Given I am on the review page,

when I rate the freelancer on all criteria, write a comment in the review form, and click "Submit,"

then my review should be sent for moderation and appear on the freelancer's profile page after approval.

• Enforce Review Length Requirements:

a. Given I am writing a review,

when I enter a review between 10 and 500 characters in length,

then the system accepts the review and allows me to submit it for moderation.

b. Given I am writing a review,

when I enter less than 10 characters,

then the system displays an error message indicating that the review must be at least 10 characters long.

c. Given I am writing a review,

when I enter more than 500 characters,

then the system displays an error message indicating that the review must not exceed 500 characters.

• Handle Empty Required Fields:

Given I am on the review page,

when I leave any required fields empty,

then the system highlights the empty fields and displays an error message.

• Cancel Review Without Saving:

Given I am on the review page,

when I click the "Cancel" button,

then the review window closes without saving any entered data.

[KEKO24-5] <u>A</u>	dmin Ability to Create New Categories Created: 16/Dec/24 Updated: 18/Jan/25
Status:	To Do
Project:	<u>Keko 2024</u>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Maryam Agaiby	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0000m:

User Story:

As an admin, I want to create a new category so that I can organize content more effectively.

Acceptance Criteria:

1. Creating a New Category

- o Given I am logged in as an admin,
- o When I navigate to the category management page,
- o **Then** I should see an option to create a new category and a form to input the category name and description.

2. Successful Category Creation

- o Given I have entered a valid category name and description in the form,
- o When I click the "Create" button,
- Then the new category should be added to the list of categories, and I should see a success message.

3. **Duplicate Category Name**

- o Given I attempt to create a category with a name that already exists,
- o When I click "Create",
- Then I should see an error message indicating that the category name is already in use.

4. Input Validation

- o Given I submit the form without entering a name or description,
- o When I click "Create",

0	Then I should see a	ııı error message	e prompting me	to provide the	required 1

[KEKO24-3] login page Created: 16/Dec/24 Updated: 16/Dec/24 Resolved: 16/Dec/24		
Status:	Done	
Project:	<u>Keko 2024</u>	
Components:	None	
Affects versions:	None	
Fix versions:	None	

Type:	Task	Priority:	Medium
Reporter:	Abdul Sami	Assignee:	Unassigned
Resolution:	Done	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	
Rank:	0 i0000t:

Generated at Sat Jun 07 15:34:42 UTC 2025 by Kseniia Kochetova using Jira 1001.0.0-SNAPSHOT#100283-rev:a18cab9d77dc93e51e1c850667233952bc1c5365.