

---

---

# ECE 651 – Foundations of Software Engineering

## Final Project Review

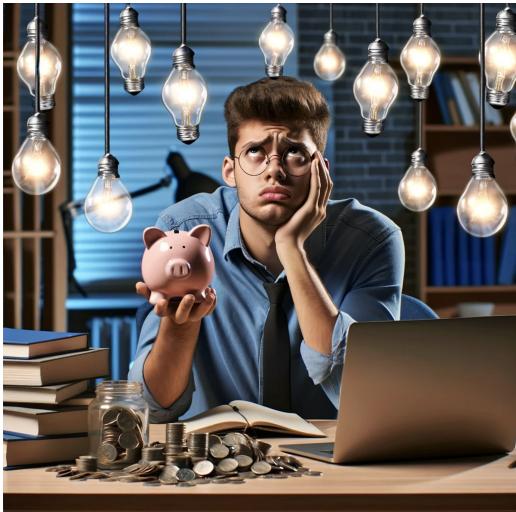
Group 7: Gehao Yu, Lucian Zhao, Pengyu Yin, Shiyu Liu, Guanjie Wang, Zekun Zhu

---

---

# Introduction / Background

University students brim with innovative ideas but often lack the resources to bring these ideas to life. Cultivate a culture of innovation and collaboration among Canadian university students through a crowdfunding platform.



**GIX Fund** aims to empower students by providing a platform where they can showcase their brilliant ideas and projects to the world and receive financial support from backers who believe in their potential.

# Project Objectives

GIX FUND

**Idea Submission:** Students can submit their projects or ideas in various categories, including technology, arts, social sciences, and more. Each project will have a detailed description, goals, and the funding amount required, broken down into manageable tiers of \$20, \$50, and \$100.

**Community Backing:** The GIXfund community, comprising students, alumni, and innovation enthusiasts, can browse through these ideas and choose to back them with contributions. This not only provides the necessary funds but also builds a network of supporters.

**Milestone Achievements:** As projects receive funding, creators can update their backers with progress, further engaging their community and building momentum towards their goals.

**Hot Tags:** Projects can be tagged with current trending topics and technologies, making it easier for backers to find and support projects that align with the latest interests and cutting-edge developments. This feature keeps the platform dynamic and aligned with the evolving interests of the community.

**Star Creators and Funders:** Recognizing and rewarding the most active and impactful members of the GIXfund community. Star creators are students whose projects have consistently met their goals and delivered impact, while star funders are those who have significantly contributed to the success of various projects.

# Deliverables - User Interface Design

The wireframe illustrates the user interface design for a crowdfunding platform, featuring several key screens:

- Homepage:** Shows a navigation bar with "YOUR OWN CROWDFUNDING PLATFORM" and "THE INSPIRATION KING SOUL PAINTER GREEN GARDEN". Below this are sections for "ALL PROJECTS" (with "TEST11" shown) and "RECOMMEND FOR YOU" (with "Figure" projects). A "FEATURED PROJECTS" section displays three projects: "Figure", "Figure", and "Figure".
- Project Detail Page (TEST11):** Displays the project title "TEST11", a thumbnail image, a reward amount of "\$10,15.20 REWARD", and a progress bar indicating "100% Funded". Below this is a "PROJECT DESCRIPTION" section with detailed text about the project's purpose and funding goals.
- Related Projects:** A sidebar listing "RELATED PROJECTS" such as "Figure", "Figure", and "Figure".
- Sign In Page:** Contains fields for "Email" and "Password", and a "SIGN IN" button.
- Sign Up Page:** Contains fields for "Email", "Password", and "Confirm Password", and a "SIGN UP" button.
- Edit Personal Info Page:** Contains fields for "Name", "Email", "Phone", and "Address", and a "Save" button.

Curved lines connect the "Figure" projects from the homepage to their respective detail pages, and from the detail pages to the "SIGN IN" and "SIGN UP" pages, indicating the flow of user interaction between these components.

# Version Control System - Git

1st main	exec07.html	
Apr 12, 2024	<a href="#">Author</a> <input type="text" value="Search message"/> <a href="#">Go</a>	
 <a href="#">Yannan</a> for all homepage components tests and fix the search not refresh bugs	<a href="#">284640dc</a> <a href="#">Open</a>	
Mar 28, 2024		
 <a href="#">Yannan</a> <a href="#">Zhou</a> authored 4 hours ago	<a href="#">17610f65</a> <a href="#">Open</a>	
 <a href="#">yannan</a> <a href="#">test</a> fixed bug fixed	<a href="#">8620f677</a> <a href="#">Open</a>	
 <a href="#">Pengyu</a> <a href="#">Yin</a> authored 5 days ago		
Mar 27, 2024		
 <a href="#">Yannan</a> <a href="#">Zhou</a> authored 6 days ago	<a href="#">1840703e</a> <a href="#">Open</a>	
 <a href="#">unit</a> <a href="#">tests</a>	<a href="#">84f00158</a> <a href="#">Open</a>	
 <a href="#">Lucifer</a> <a href="#">Zhou</a> authored 6 days ago		
 <a href="#">fix the issue related to project test</a>	<a href="#">4993c153</a> <a href="#">Open</a>	
 <a href="#">O-Quarantine</a> <a href="#">Guo</a> authored 6 days ago		
 <a href="#">fix the merge issue for the delete project api</a>	<a href="#">28464088</a> <a href="#">Open</a>	
 <a href="#">O-Quarantine</a> <a href="#">Guo</a> authored 6 days ago		
 <a href="#">fix the merge issue for the delete project api</a>	<a href="#">1fbfe1c7</a> <a href="#">Open</a>	
 <a href="#">O-Quarantine</a> <a href="#">Guo</a> authored 6 days ago		
Mar 26, 2024		
 <a href="#">Merge commit</a> <a href="#">zg2275d0</a> into adminFrontend		
 <a href="#">Lucian</a> <a href="#">Zhao</a> authored 1 week ago		
 <a href="#">admin</a> <a href="#">update</a>		
 <a href="#">Lucian</a> <a href="#">Zhao</a> authored 1 week ago		
 <a href="#">12441 Add a ten and dialog for admin to delete projects</a>		
 <a href="#">Zekun</a> <a href="#">Zhou</a> authored 1 week ago		
 <a href="#">Merge commit</a> <a href="#">zg2275d0</a> into origin/User-Backend-Code		
 <a href="#">Lucian</a> <a href="#">Zhao</a> authored 1 week ago		
 <a href="#">Modify project related project handlers.</a>		
 <a href="#">Gehao</a> <a href="#">Yu</a> authored 1 week ago		
 <a href="#">tmp</a> <a href="#">admin</a> <a href="#">show</a> <a href="#">projects</a>		
 <a href="#">Lucien</a> <a href="#">Zhao</a> authored 1 week ago		
 <a href="#">admin</a> <a href="#">redirection</a> <a href="#">and</a> <a href="#">sign-in</a>		
 <a href="#">Lucien</a> <a href="#">Zhao</a> authored 1 week ago		
 <a href="#">[za24] fix donate overflow</a>		
 <a href="#">Zekun</a> <a href="#">Zhou</a> authored 1 week ago		
 <a href="#">[za24] fix donate problem</a>		
 <a href="#">Zekun</a> <a href="#">Zhou</a> authored 1 week ago		
 <a href="#">Guardian</a> <a href="#">Vibes</a> authored 6 days ago		
 <a href="#">Guardian</a> <a href="#">Vibes</a> authored 6 days ago		
 <a href="#">participated</a> <a href="#">done</a>		
 <a href="#">Pengyu</a> <a href="#">Yin</a> authored 1 week ago		
 <a href="#">participated</a> <a href="#">done</a>		
 <a href="#">Pengyu</a> <a href="#">Yin</a> authored 1 week ago		



# Prototype

- Think out ideas and design the basic layout of the project.
- Prepare for the later implementation.
- UML UI design
- Developing environment set-up
- All basic function modules set-ups

Completed Issues (closed)	
Create the Design Document for User related Page Backend	9
#13 <span>server</span> 	
Create the Design Document for Home Page Backend	
#12 <span>server</span> 	
Create the Design Document for Project Page Backend	
#11 <span>server</span> 	
Set up Backend Dev environment for User Related Page	
#10 <span>server</span> 	
Set up Backend Dev environment for Home Page	
#9 <span>server</span>	
Set up Backend Dev environment	
#8 <span>server</span> 	
Client: Project Framework Setup	
#4 <span>Client</span> 	
Client: UI Design	
#3 <span>Client</span> 	
Client: UML design	
#2 <span>Client</span> 	

# Version1

Implement the first version of the project. It includes:

- Correct pages and references.
- Database setup.
- Backend setup.
- Basic frontend - backend interactions.

Completed Issues (closed)	7
Client: Complete project list page	#19 Client 🌐
Client: Prepare main page components and finish bar component	#18 Client 🌐
Client: Complete coding main page and its corresponding content	#17 Client 🌐
Client: Improve profile page content	#15 Client 🌐
Client: Code UI Layout(Profilepage)	#7 Client 🌐
Client: Code UI Layout(Projectpage)	#6 Client 🌐
Client: Code UI Layout(Homepage)	#5 Client 🌐

# Version2

Implement the second version of the project.

It includes:

- Full frontend - backend interactions.
- Correct responses and enough fake resources.
- Basic test scripts.
- Improvement plan and schedule.

Completed Issues (closed)	
#29	Client 
Client: Write full interactions profile page	
#28	Client 
Client: Write full interactions main page	
#27	Client 
Client: Write full interactions project page	
#26	Client 
Prepare review documents	
#25	Client 
Client: Write basic interactions project-page	
#24	Client 
Client: Write basic interactions profile-page	
#23	Client 
Client: Write basic interactions main-page	

# Version 3

Implement the final version of the project. It includes:

- Complete full interactions.
- Complete full testing.
- Complete improvements to user experience.

Completed Issues (closed)	
	□ 6
Client: Finish profile page view by others	
#35	Client 
Client: Create fake projects	
#34	Client 
Search projects	
#33	server 
Client: Finish the following jobs	
#32	Client 
Client: Finish the following jobs	
#31	Client 
Client: Finish the following jobs	
#30	Client 

# Agile Project Management

## Kanban Board

- Set up our Kanban in the issue board of GitLab
- Divided it into groups to reflect the workflow stages.

## User Stories and Sprints

- “Product owner” created each issue for user stories by clearly defining the scope and acceptance criteria in the description.
- We also used labels to categorize issues (e.g., Client, Server) and milestones to group issues into sprints or phases.

## Collaboration and Progress Tracking

- Scrum Master is responsible for assigning tasks to others.
- Developers can update the status of tasks and communicate by comments.

Open	Closed
Create the Design Document for Home Page Backend server #12 Feb 4	Set up Backend Dev environment for User Related Page server #10 Feb 4
Create the Design Document for User related Page Backend server #13 Feb 4	Client: Code UI Layout(Homepage) Client #5 Feb 7
Client: Improve profile page content Client #15 Feb 14	Client: UI Design Client #3 Jan 31
Client: Complete coding main page and its corresponding content Client #17 Feb 14	Client: Code UI Layout(Profilepage) Client #6 Feb 7
Client: Check main page content related data and communicate with backend Client #20 Feb 21	Client: Code UI Layout(Profilepage) Client #7 Feb 7
Client: Write basic interactions main-page Client #23 Feb 21	Create the Design Document for Project Page Backend server #11 Feb 4
Prepare review documents Client #26 Tomorrow	Purpose of the nested Express app at server Client #14
	Set up Backend Dev environment server #8 Feb 4
	Client: Project Framework Setup Client #4 Jan 31

# Team Communication Following Scrum

## Stand-up Meeting

- Tools: WeChat
- Cover: What we did? What are impediments? What we will do next?

## Weekly Sprint Meeting: Offline & Online

- Schedule: Our meetings are held offline every Wednesday after class, with each meeting lasting from 30 minutes to about an hour.
- Cover:
  - Product demo - To exhibit accomplished tasks
  - User stories - To confirm complete stories and discuss how to handle incomplete ones
  - Product backlog – To assess and adjust the product backlog dynamically
- Additional meetings: we will hold on discord.

## Meeting Records:

- [https://docs.google.com/document/d/1TK\\_GmIRsVEgQEtvQkEXJd6c675XKYJW1pNZINXAGZU4/edit](https://docs.google.com/document/d/1TK_GmIRsVEgQEtvQkEXJd6c675XKYJW1pNZINXAGZU4/edit)

# Issues and PRs

After completing work on your own branch, list the details of the changes they made and submit a PR request.

At least one team member serves as a reviewer, reading or video chatting questions. After review and approval. The commit will be merged into the master branch

A screenshot of a web interface for managing issues and pull requests. At the top, there are navigation tabs: Open (1), Merged (6), Closed (2), and All (9). Below the tabs is a search bar with 'Recent searches' and a dropdown for 'Search or filter results...'. A 'Bulk edit' button and a 'New merge request' button are also present. The main area displays a list of items, each with a title, creation date, status (e.g., Approved, Merged, Closed), and update information. Some items have labels like 'Version 3' or 'server'. The items listed include:

- Change all issues and bugs from main page. (Approved, updated 2 days ago)
- Update the unit tests for project apis. (Merged, updated 6 days ago)
- Update the unit test for Project backend api. (Closed, updated 6 days ago)
- Merge User backend unit tests to main. (Merged, updated 1 week ago)
- Homepage dev. (Merged, updated 1 week ago)
- Merged new handlers for UserAuth and UserProfile and Admin. (Merged, updated 2 weeks ago)
- Merge of user auth schema to main. (Merged, updated 3 weeks ago)
- Merge project to main. (Closed, updated 6 days ago)
- Merge Joint Debugging rbranch codes to Main. (Merged, updated 1 month ago)

At the bottom left, there are two red buttons labeled 'Client' and 'server'.

A screenshot of a Jira-style issue detail page. At the top, it says 'Client: Check project page content related data and communicate with backend'. Below that is an 'Edit' button and a three-dot menu. The main content area shows an issue summary: 'Issue created 1 month ago by Zekun Zhu'. It includes a file upload section with a placeholder 'Drag your designs here or click to upload.', a 'Child items' section (0), and a 'Linked items' section (0). The 'Activity' section lists the following history:

- Zekun Zhu changed due date to February 21, 2024 1 month ago
- Zekun Zhu changed milestone to %Version2 1 month ago
- Zekun Zhu added 'client' label 1 month ago
- Zekun Zhu assigned to @z93zhu 1 month ago
- Zekun Zhu closed 1 month ago

A second screenshot of a Jira-style issue detail page. At the top, it says 'Search projects'. Below that is an 'Edit' button and a three-dot menu. The main content area shows an issue summary: 'Issue created 2 weeks ago by Lucian Zhao'. It includes a file upload section with a placeholder 'Drag your designs here or click to upload.', a 'Child items' section (0), and a 'Linked items' section (0). The 'Activity' section lists the following history:

- Lucian Zhao changed due date to March 18, 2024 2 weeks ago
- Lucian Zhao changed milestone to %Version 3 2 weeks ago
- Lucian Zhao added 'server' label 2 weeks ago
- Lucian Zhao assigned to @322has 2 weeks ago
- Zekun Zhu closed 58 minutes ago

# Contributions

## Front-End

### Zhu Zekun - Project Page

Developed project display and detail pages, enhancing user browsing experience.

Integrated dynamic loading and interaction of project data

### Yin Pengyu - User Profile Page

Designed and implemented the UI for the user profile page  
Integrated frontend with backend for dynamic data binding and display

### Liu Shiyu - Home Page

Responsible for the layout design and frontend logic of the homepage  
Implemented real-time information display and user interaction features

## Back-End

### Wang Guanjie - Backend Integration for Project Page

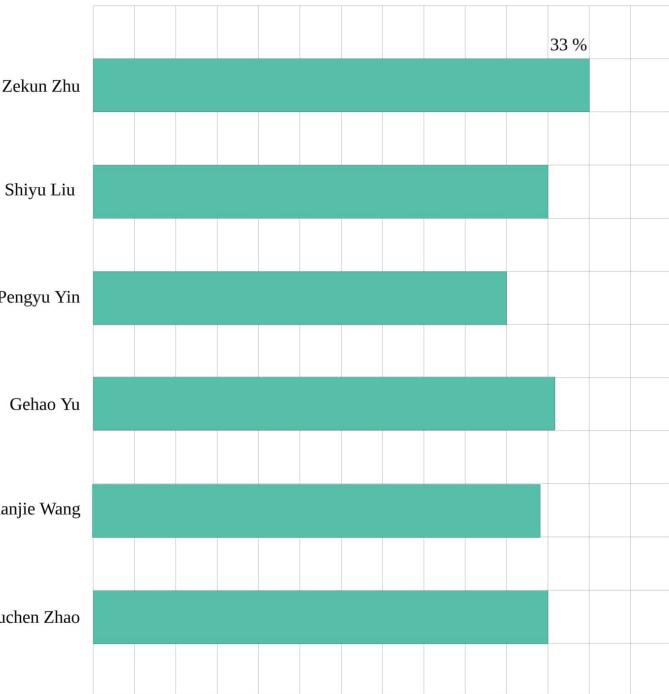
Handled backend logic and database design for project information  
Provided efficient data interfaces for frontend consumption

### Yu Gehao - Backend Integration for User Authentication, Profile Page and Admin User

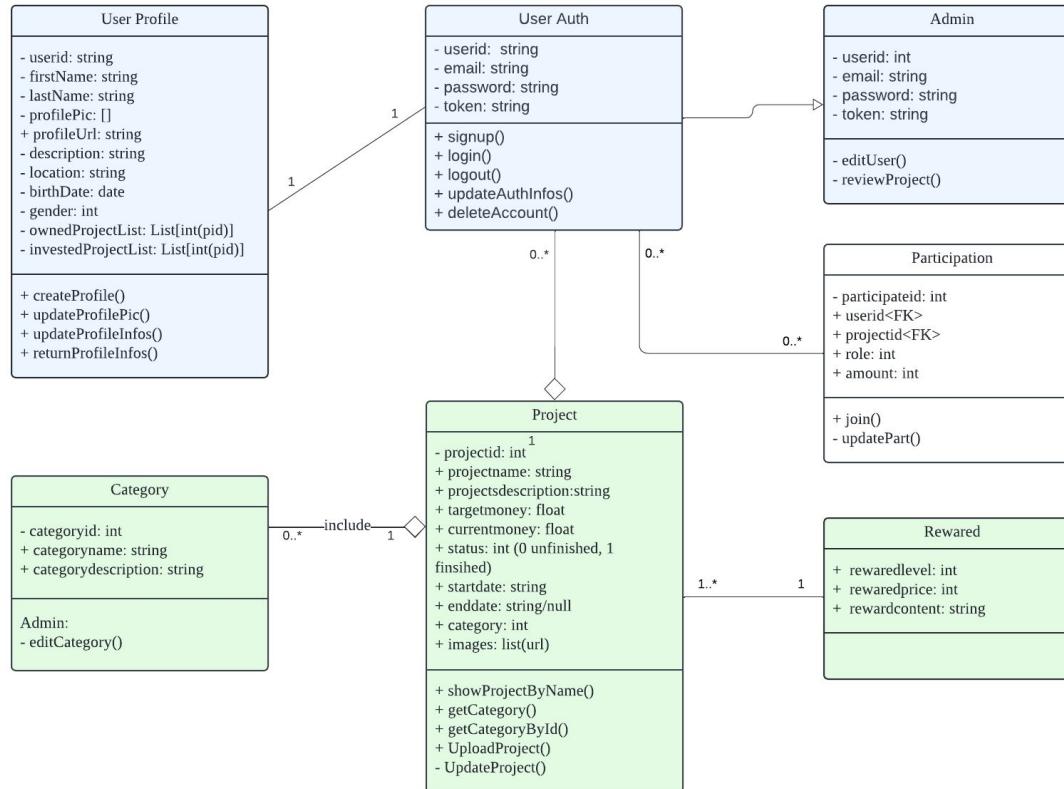
Developed data processing and storage logic for user profiles, authentication and admin user  
Provided APIs required by the frontend

### Zhao Lucian - Backend Integration for Home Page, and Administration

Developed backend services for data display on the homepage  
Ensured efficient and secure data transmission



# UML and Dataflow



# Unit Testing: Jest Library for BackEnd

## GET /showProjectBy endpoint

- ✓ should return projects matching the category ID
- ✓ should return projects matching the projectname
- ✓ should return projects matching the pid

## POST /showUserProject

- ✓ should respond with user projects if present 77 ms
- ✓ should respond with 404 if user has no projects 10 ms
- ✓ should respond with 500 if no token provided 7 ms

## GET /getCategoryInfoById

- ✓ should return category info matching the category ID 23 ms
- ✓ should return 400 if category does not exist 22 ms

## POST /showUserDonateProject

- ✓ Should return 401 if no token provided 43 ms
- ✓ Should return 400 if user has not participated in any p 30 ms
- ✓ Should pass 7 ms

## POST /deleteproject

- ✓ Should return 400 if no token provided 47 ms
- ✓ Should pass 7 ms

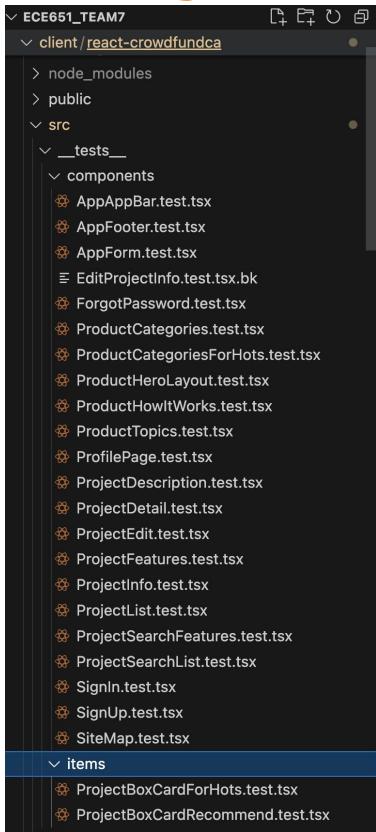
## GET /getCategorys

- ✓ should return category info matching the category ID 22 ms

## project.test.js

- ✓ POST /uploadProject
  - ✓ should successfully upload a project 66 ms
  - ✓ should respond minimally 9 ms

# Unit Testing: React Testing Library



A screenshot of a file explorer window titled "ECE651\_TEAM7". The "client/react-crowdfundca" folder is expanded, showing the "src" directory which contains "components", "tests", and "items" sub-directories. The "tests" directory is expanded, displaying numerous test files such as "AppBar.test.tsx", "AppFooter.test.tsx", "AppForm.test.tsx", "EditProjectInfo.test.tsx.bk", "ForgotPassword.test.tsx", "ProductCategories.test.tsx", "ProductCategoriesForHots.test.tsx", "ProductHeroLayout.test.tsx", "ProductHowItWorks.test.tsx", "ProductTopics.test.tsx", "ProfilePage.test.tsx", "ProjectDescription.test.tsx", "ProjectDetail.test.tsx", "ProjectEdit.test.tsx", "ProjectFeatures.test.tsx", "ProjectInfo.test.tsx", "ProjectList.test.tsx", "ProjectSearchFeatures.test.tsx", "ProjectSearchList.test.tsx", "SignIn.test.tsx", "SignUp.test.tsx", and "SiteMap.test.tsx". A blue bar at the bottom highlights the "items" folder.



File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
EmptyProjectCard.js	100	100	100	100	
ForgotPassword.tsx	69.23	64.28	80	69.23	21-23,31
ProjectDetail.tsx	100	100	100	100	
ProjectEdit.tsx	100	100	100	100	
ProjectList.tsx	100	100	100	100	
ProjectSearchList.tsx	100	100	100	100	
SignIn.tsx	50	75	50	50	39,49,58-95
theme.ts	100	100	100	100	
withRoot.tsx	100	100	100	100	
theme.ts	100	100	100	100	
withRoot.tsx	100	100	100	100	
Button.tsx	100	75	100	100	15
CircleProgressWithLabel.tsx	100	0	100	100	40
defer.tsx	100	100	100	100	
validation.ts	100	87.5	100	100	14
src/components/items	63.59	61.81	52.38	62.6	
AlertDialogWithBtn.tsx	60	75	25	60	25,29,33-34
AppBar.tsx	100	100	100	100	
PageContentWithEditor.tsx	90.9	100	75	90.9	143
Paper.tsx	100	80	100	100	14
ProgressBar.tsx	100	100	100	100	
PageContentWithEditor.tsx	90.9	100	75	90.9	143
Paper.tsx	100	80	100	100	14
ProgressBar.tsx	100	100	100	100	
TextField.tsx	100	100	100	100	
Toolbar.tsx	100	100	100	100	
Typography.tsx	100	100	100	100	
AppFooter.tsx	100	100	100	100	
AppForm.tsx	100	100	100	100	
ProductCategories.tsx	83.33	50	66.66	90.9	32
ProductCategoriesForHots.tsx	81.81	50	66.66	90	30
ProjectSearchFeatures.tsx	100	100	100	100	
SiteMap.tsx	100	100	100	100	

**RUNS** src/\_\_tests\_\_/pages/ProfilePage.test.tsx

**Test Suites:** 0 of 1 total  
**Tests:** 2 passed, 2 total  
**Snapshots:** 0 total  
**Time:** 1 s, estimated 2 s

# Integration Testing: Back-end handler Testings

```
describe('For endpoints:', () => {
  it('/register should create a new user and return status 201', async () :Promise<void> => {
    const userData :{} = {
      email: 'test@example.com',
      password: 'password',
      firstName: 'Test',
      lastName: 'User',
    };

    const response = await request(app)
      .post('/api/auth/register')
      .send(userData);

    expect(response.statusCode).toBe(expected: 201);
    expect(response.text).toBe(expected: 'User created with profile');
    // Add more assertions as needed
  });

  it('/login authenticates a user and returns a token', async () :Promise<void> => {
    // First, create a user directly in the database
    const hashedPassword = await bcrypt.hash('password123', 10);
    await UserAuth.create(docs: {
      email: 'testlogin@example.com',
      password: hashedPassword,
    });
  });
});
```

Test Results		303 ms
✓	profile.test.js	303 ms
✓	Profile Routes	303 ms
✓	Profile management	303 ms
✓	/updateProfileInfos updates user profile information	186 ms
✓	/getUserProfile retrieves the authenticated user's profile	117 ms

➤	📁 coverage
➤	📁 models 7 files, 82% lines covered
JS	AdminSchema.js 100% lines covered
JS	projectCategorySchema.js 100% lines covered
JS	projectDbUtil.js 70% lines covered
JS	projectRewardedSchema.js 100% lines covered
JS	projectSchemas.js 100% lines covered
JS	UserAdminDBUtil.js
JS	UserAuthSchema.js 100% lines covered
JS	UserProfileSchema.js 100% lines covered
JS	userSchemas.js
➤	📁 node_modules /library root

➤	📁 Test Results	640 ms
	auth.test.js	640 ms
	Auth Routes	640 ms
	For endpoints:	640 ms
	/register should create a new user and return status 201	117 ms
	/login authenticates a user and returns a token	104 ms
	/getUserAuth returns user auth info for a logged-in user	106 ms
	/updateUserAuth updates user info for logged-in user	205 ms
	/deleteAccount deletes the user account for logged-in user	104 ms
	/logout confirms user logout	4 ms

# Integration Testing: Postman

- Test before integration

Project	
-	projectid: int
+	projectname: string
+	projectsdescription:string
+	targetmoney: float
+	currentmoney: float
+	status: int (0 unfinished, 1 finisched)
+	startdate: string
+	enddate: string/null
+	category: int
+	images: list(url)
+ showProjectByName()	
+	getCategory()
+	getCategoryById()
+	UploadProject()
-	UpdateProject()

The screenshot shows the Postman interface with a POST request to `localhost:3001/api/project/uploadProject`. The request body is filled with the following data:

Key	Value	Description
projectname	Boats	
projectsdescription	Original oil painting "Boats in the Bay", ...	
targetmoney	451	
currentmoney	15	
enddate	2025/10/25	
startdate	2024/2/25	
categoryId	2	
userId	1	
token	csdfasfas	
rewardlevel	1	
rewardprice	100	
rewardcontent	one NFT picture	

The response shows a JSON object with the following content:

```
{
  "success": true,
  "message": "Project saved to database",
  "project": {
    "pid": "d3016e8b131fc9a7177190c0c832de1250555fa6ec27cc576d021fc15cf2",
    "token": "csdfasfas",
    "projectname": "FakeBoats2",
    "projectsdescription": "Original oil painting 'Boats in the Bay'. Artwork captures the beauty of Balikova Bay, with colorful boats and yachts dotting the serene waters. Painted with brush strokes and use of a palette knife, this artwork brings a dynamic and textured feel to the canvas. This one-of-a-kind work is not a print or reproduction, ensuring that you are adding a truly unique piece to your art collection.\n\nReady to Hang: Shipped on canvas with painted sides, allowing for immediate display without the need for additional framing.\n\nMaterials: Created with high-quality oil paints, ensuring vibrant colors and enduring beauty.\n\nAuthentication: Personally signed on the front and accompanied by a Certificate of Authenticity, ensuring the uniqueness and origin of this piece.\n\nNote: I occasionally paint the sides before shipping, so they may not appear painted (look white) in the photo or video. Please disregard this discrepancy, as it does not affect the overall presentation.\n\nShipping: Uninterrupted stress-free delivery experience with Worldwide Express Mail Service. Each artwork is carefully packaged using thick foam plastic and sturdy cardboard to ensure maximum protection during transit.\n\nUpdate 2024: Despite the situation in Israel, I continue to work and ship paintings internationally. Delivery services operate efficiently, ensuring timely and reliable shipping.\n\nCustoms and Import Taxes: Buyers are responsible for any customs duties and import taxes that may apply once the order arrives in the destination country. I'm not responsible for delays caused by customs procedures. Please be aware of your country's import regulations to ensure a smooth delivery process.\n\nFeel free to contact me for any questions or additional information."
  },
  "categoryId": "2",
  "images": [
    "1708992880230_2.1.png",
    "1708992880239_2.2.png"
  ],
  "userId": "1",
  "targetmoney": "400",
  "currentmoney": "15",
  "startdate": "2024-02-25T09:00:00Z",
  "rewardlevel": [
    ...
  ]
}
```

**Questions ?**