American International University-Bangladesh (AIUB)

Department of Computer Science Faculty of Science & Technology (FST)

"StoreBhai: A Peer-to-Peer Storage Rental Platform for Bangladesh"

A Software Engineering Project

Submitted By

Semester : Summer 24-25		**Section:B**	**Group Number:08**	
SL	**Student Name**	**Student ID**	**Contribution (CO3 + CO4)**	**Individual Marks**
1	SABBIR AHMED	22-49109-3	25%	
2	ZOBAYER AL MAHMUD	22-47989-2	30%	
3	MD.MUSTAKIM ABIR	22-47645-2	25%	
4	SADIA RAHAMAN	21-45900-3	20%	

The project will be evaluated for the following Course Outcomes

CO3 (PO-g-1) ***Select appropriate software engineering models, project management roles and their associated skills for the complex software engineering project and evaluate the sustainability of developed software, taking into consideration the societal and environmental aspects***	Total Marks	
Selection of Software Engineering Models: Process model selection and presents sufficient evidence to support argument for the model selection	[5 Marks]	
Role identification and Responsibility Allocation: Well-planned project with proper role identification and responsibility allocation in the project management activities	[5Marks]	
Formatting and Submission: Submission, Defense, Completeness, Spelling, grammar, and Organization of the Project report	[5Marks]	
CO4 (PO-k-1)	Total Marks	

Apply engineering management principles and economic decision making to develop software engineering project management plan.		
Project WBS and Testcases: Relevant WBS (project task list) and testcases for the proposed project are stated properly.	[5Marks]	
Effort Estimation and Scheduling: Project estimation was described using proper effort estimation or schedules based on available project resources	[5Marks]	
Risk Management: Sufficient and appropriate risks are identified, analyzed, and properly categorized or prioritized.	[5Marks]	

Background

The idea behind this project stems from the growing need for affordable and secure storage solutions. Many people, especially business owners, students, and drivers, require temporary storage for their belongings but often struggle to find reliable options. Additionally, people with unused space in their homes, offices, or garages have no way of monetizing that space. Existing solutions in Bangladesh don't address these specific needs. This project aims to fill that gap by creating a platform where people can rent out unused storage space and find storage when needed**.**

Scope and Features:

The goal of this project is to build a web platform called StoreBhai, which will connect people looking for storage (renters) with those who have unused space available (hosts).

The platform will offer features such as:

- A signup and login system for both renters and hosts.
- A listing function for the host to list their unused spaces.
- A search function for renters to find available storage spaces.
- Booking and payment integration for securing storage.
- Profile management for users to update personal details.
- An admin panel for managing users, listings, bookings, payments, and disputes.

User Story Table:

As a	I want to	So that	Acceptance Criteria
Host	List my unused storage space	Renters can find and book my space	Must be able to add space, provide details, and update availability status.
Renter	Search for storage spaces by location	Find a space nearby for temporary storage	Search results must show accurate locations and availability.
Admin	Manage users and bookings	Ensure the platform is running smoothly	Admin can view, edit, and remove users, and manage bookings.
Admin	Process payments for bookings	Ensure payments are secure and tracked	Payments must be processed securely, and receipts sent to the renter.

Software Development Process Model Selection: Scrum

For the Store Bhai project, we have selected Scrum as our software development process model. Scrum is highly suitable for projects where requirements are expected to evolve, as it provides both structure and flexibility through iterative development and continuous stakeholder feedback. Store Bhai operates in a dynamic retail environment, where customer expectations, payment methods, and shop management features can frequently change. The Scrum framework, with its fixed 90-day Sprints, allows the team to adapt to these changes efficiently, ensuring that the product evolves in direct response to user feedback and market demands.

Our development team benefits from Scrum's defined roles (Product Owner, Scrum Master, Development Team), which establish clarity in responsibilities and improve communication. Daily standups, sprint reviews, and retrospectives within each 90-day cycle ensure that blockers are quickly addressed, collaboration is strengthened, and progress remains aligned with business goals. With a clear Product Backlog and Sprint Backlog, tasks are prioritized effectively to meet Store Bhai's objectives.

Scrum is flexible in adapting to changes in scope, technology, or user requirements. If new features such as loyalty programs or advanced reporting are introduced, they can be added to the Product Backlog and implemented in subsequent 90-day sprints without disrupting current progress. This adaptability makes Scrum well-suited for projects that depend heavily on customer feedback and continuous improvement.

Store Bhai provides a creative solution to a real-life problem by replacing fragmented manual systems with a digital, centralized retail management platform. This enhances transparency, reduces errors, and improves customer satisfaction. The target users include small and medium shop owners, employees, and customers, all of whom benefit from simplified store operations and improved efficiency.

The project also contributes to scientific and professional development by documenting iterative improvements, testing cycles, and backlog refinements. Compared to Extreme Programming (XP), Scrum offers more formal structure in terms of ceremonies and roles, which suits our team better. While XP emphasizes engineering practices such as pair programming and collective code ownership, Scrum's focus on process management, stakeholder involvement, and sprint-based delivery makes it more aligned with Store Bhai's goal of delivering a robust and user-validated product incrementally.

Scrum effectively manages project risks by breaking work into smaller increments and ensuring that risks are identified early within each 90-day sprint. Regular reviews and retrospectives reduce uncertainty, while timely delivery of increments ensures deadlines are met. Overall, Scrum provides the right balance of flexibility, structure, and collaboration, making it more suitable for Store Bhai than XP or other traditional models**.**

1. Project Role Identification and Responsibilities

Main Roles in the Project

For the Store Bhai project, several roles cover both development and management aspects. Since the project is handled by a small team, members often take on multiple responsibilities**. The Main roles are:**

1. **Product Owner – Sabbir**
2. **Scrum Master – Zobayer**
3. **Development Team – Abir and Sadia**

Responsibilities in Key Stages

Stage	Roles & Responsibilities
Requirements Gathering	Sabbir gathers business requirements from shop owners and customers. Zobayer, Abir, and Sadia assist in backlog refinement.
System Design	Abir and Sadia design architecture and user interface. Zobayer ensures collaboration and adherence to Scrum practices.

Stage	Roles & Responsibilities
Implementation	Abir and Sadia develop features iteratively during sprints, delivering increments of working software.
Testing	Abir and Sadia conduct functional and user acceptance testing within each sprint to ensure quality.
Deployment	Abir and Sadia manage deployment of increments, reviewed during Sprint Review with stakeholders.

Decision-Making, Quality Assurance, and Resource Management

- Decision-Making: Shared among Sabbir (business priorities), Zobayer (process), and Abir & Sadia (technical solutions).
- Quality Assurance: Ensured by Abir & Sadia with continuous testing. Zobayer ensures compliance with Scrum principles.
- Resource Management: Coordinated by Zobayer, with alignment to business priorities ensured by Sabbir.

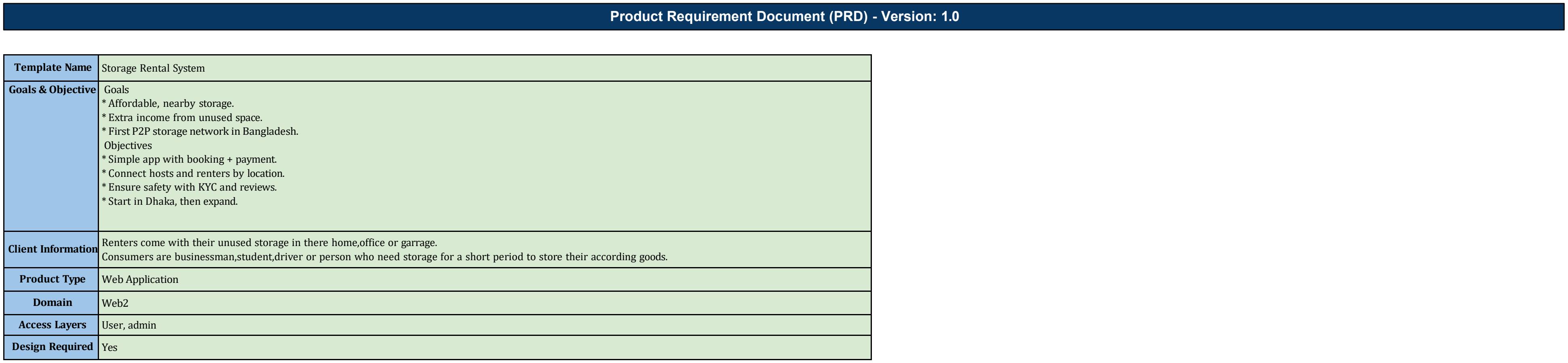
Distribution of Responsibilities and Justification

Responsibilities are distributed according to Scrum’s defined roles:

- Sabbir ensures that Store Bhai’s business needs are met by prioritizing features in the backlog.
- Zobayer ensures that the Scrum process is followed, facilitating communication and removing impediments.
- Abir and Sadia handle technical work (design, coding, testing, deployment) and collaborate closely to deliver working increments.

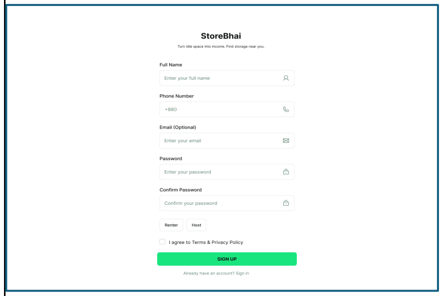
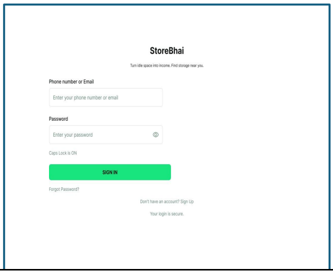
This balanced distribution ensures smooth project progress, timely delivery, and a high-quality solution that benefits Store Bhai’s stakeholders.

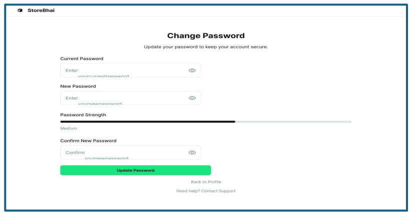
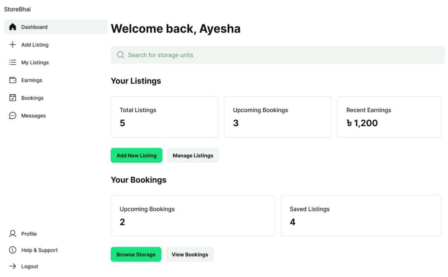
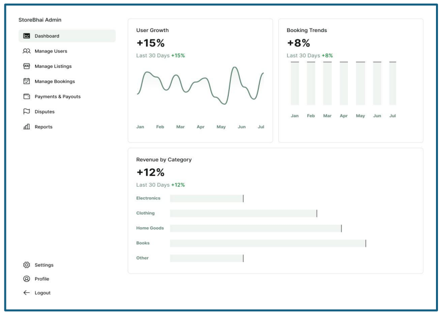
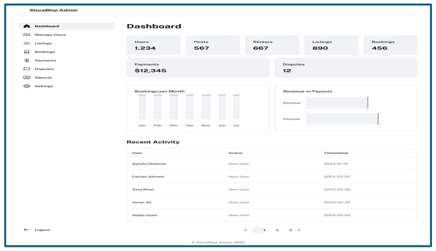
SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) / PRODUCT REQUIREMENTS DOCUMENT (PRD)

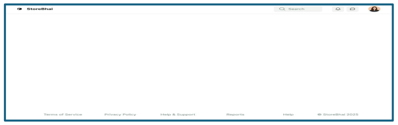
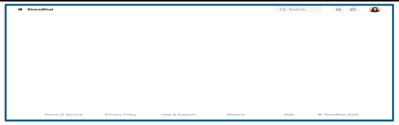
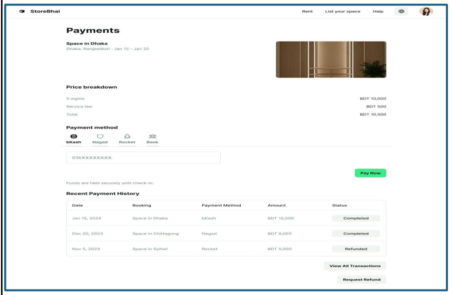
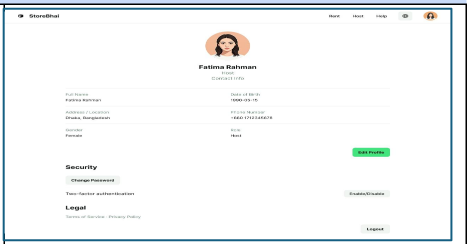
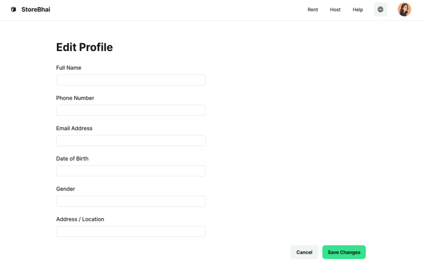


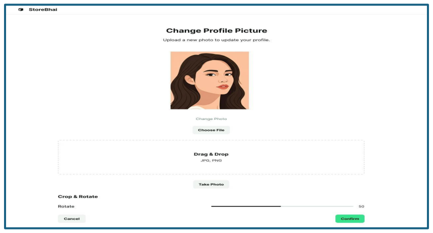
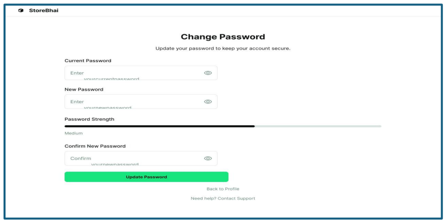
Product Requirement Document (PRD) - Version: 1.0

Template Name	Storage Rental System
Goals & Objective	Goals * Affordable, nearby storage. * Extra income from unused space. * First P2P storage network in Bangladesh. Objectives * Simple app with booking + payment. * Connect hosts and renters by location. * Ensure safety with KYC and reviews. * Start in Dhaka, then expand.
Client Information	Renters come with their unused storage in there home,office or garrage. Consumers are businessman,student,driver or person who need storage for a short period to store their according goods.
Product Type	Web Application
Domain	Web2
Access Layers	User, admin
Design Required	Yes

Phase	Item No	Specification [Role]	Screen Definition	User Story	Acceptance Criteria	UI Screen / Wireframe
Feature: 1		**Auth Feature**				
phase 1	1	Signup [admin] [User]	<Form> <input = name> <input = phone/email> <input = password> <input = re-password> <select = user-type> (Host / Renter) <button = signup> <link = signin> </Form>	As a new user, I want to sign up with my details so I can create an account and start using StoreBhai. As an admin, I want to store and verify signup info so only valid accounts are created.	Phone/Email must be unique. Otherwise show “Account already exists.” Password must be at least 8 characters. Otherwise show “Password too short.” Re-entered password must match. Otherwise show “Passwords do not match.” User type must be selected (Host/Renter). Otherwise show “Select a user type.” On success → account stored in DB + confirmation SMS/email sent + redirect to Dashboard.	 StoreBhai Turn idle space into income. Find storage near you. Full Name Enter your full name Phone Number +880 Email (Optional) Enter your email Password Enter your password Confirm Password Confirm your password Renter Host □ I agree to Terms & Privacy Policy SIGN UP Already have an account? Sign in
phase 1	2	Signin [admin] [User]	<Form> <input = phone/email> <input = password> <button = signin> <link = signup> <link = forgot password> </Form>	As a registered user, I want to sign in securely so I can access my account. As an admin, I want to validate credentials so only valid accounts can log in.	Valid phone/email + password → login success → redirect to Dashboard. Wrong phone/email → show “Account not found.” Wrong password → show “Incorrect password.” Suspended account → show “Account inactive. Contact support.” On success → user logged in and session created.	 StoreBhai Turn idle space into income. Find storage near you. Phone number or Email Enter your phone number or email Password Enter your password Caps Lock is ON SIGN IN Forgot Password? Don't have an account? Sign Up Your login is secure.

phase 1	3	Forgot password[admin] [User]	<Form> <input = phone/email> <input = OTP code> <input = new password> <input = re-password> <button = reset password> </Form>	As a user, I want to reset my password with OTP so I can regain access if I forget it. As an admin, I want to ensure OTP verification before allowing password reset.	Phone/Email must exist in system. Otherwise → "Account not found." OTP code sent to phone/email. Must match input. If not → "Invalid code." OTP expires after 10 minutes. New password must meet policy (≥ 8 characters). Re-password must match. Otherwise → "Passwords do not match." On success → password updated, redirect to Signin.	 StoreBhai Change Password Update your password to keep your account secure. Current Password Enter yourcurrentpassword New Password Enter yournewpassword Password Strength Medium Confirm New Password Confirm yournewpassword Update Password Back to Profile Need help? Contact Support
Feature: 2		**Dashboard**				
phase 1	4	Menu [Host][Renter]	<Menu> - Dashboard - Search Space - My Bookings - Payments - Messages - Profile > View Profile > Edit Profile > Change Photo > Change Password - Help & Support - Logout </Menu>	As a renter, I want a clear menu so I can navigate easily to find and manage storage.	Dashboard → takes user to main summary page. Search Space → opens map/list of spaces. My Bookings → opens booking history. Payments → shows receipts/refunds. Messages → opens chat screen. Profile → goes to profile sub-pages. Logout → ends session and returns to Signin.	 StoreBhai Dashboard Add Listing My Listings Earnings Bookings Messages Profile Help & Support Logout Welcome back, Ayesha Search for storage units Your Listings Total Listings 5 Upcoming Bookings 3 Recent Earnings ৳ 1,200 Add New Listing Manage Listings Your Bookings Upcoming Bookings 2 Saved Listings 4 Browse Storage View Bookings
phase 1	5	Menu[admin]	<Menu> - Dashboard - Manage Users - Manage Listings - Manage Bookings - Payments & Payouts - Disputes - Reports - Settings - Profile > Edit Profile > Change Password - Logout </Menu>	As an admin, I want a menu with management tools so I can monitor and control the platform.	Each menu option opens correct section. Unauthorized users cannot see Admin menu. Dashboard → shows KPIs. Manage Users → shows list with filter/search. Reports → export CSV/PDF. Logout → securely ends session.	 StoreBhai Admin Dashboard Manage Users Manage Listings Manage Bookings Payments & Payouts Disputes Reports Settings Profile Logout User Growth +15% Last 30 Days +15% Jan Feb Mar Apr May Jun Jul Booking Trends +8% Last 30 Days +8% Jan Feb Mar Apr May Jun Jul Revenue by Category +12% Last 30 Days +12% Electronics Clothing Home Goods Books Other
phase 1	6	Dashboard [admin] [User]	<Dashboard> - Total Users - Active Hosts - Active Renters - Total Listings - Active Bookings - Completed Bookings - Payments & Payouts summary - Disputes pending - Reports quick link </Dashboard>	As an admin, I want to see key stats (users, bookings, payments, disputes) so I can quickly understand platform health.	Dashboard loads with updated live data. Clicking "Total Users" → goes to User Management page. Clicking "Listings" → goes to Listings Management page. Clicking "Disputes" → goes to Dispute Management page. Data should auto-refresh every X minutes or on refresh.	 StoreBhai Admin Dashboard Manage Users Listings Bookings Payments Disputes Reports Settings Logout Dashboard Users 1,234 Hosts 567 Renters 667 Listings 890 Bookings 456 Payments $12,345 Disputes 12 Bookings per Month Jan Feb Mar Apr May Jun Jul Revenue vs Payouts Revenue Payouts Recent Activity User / Action / Timestamp Ayesha Rahman / New User / 2024-01-15 Farhan Ahmed / New User / 2024-02-20 Zara Khan / New User / 2024-03-25 Imran Ali / New User / 2024-04-30 Nadia Islam / New User / 2024-05-05 < 1 2 3 > © StoreBhai Admin 2025
Feature: 3		**Top bar and Bottom Bar**				

phase 1	7	Top bar	<Top Bar> - Logo (StoreBhai / StoreBhai Admin) - Search (Users, Listings, Spaces, Bookings) - Notifications icon (alerts, disputes, bookings) - Profile icon (dropdown: Profile, Settings, Logout)	As a user, I want to quickly access search, notifications, and profile from the top bar so I can navigate faster. As an admin, I want the top bar to also include system search and alerts so I can manage StoreBhai efficiently.	Logo → redirects to Dashboard (User → User Dashboard, Admin → Admin Dashboard). Search → opens user-friendly search (User: spaces, Admin: users/listings/bookings). Notifications icon → shows unread count; opens notifications list. Profile icon → opens dropdown with Profile, Settings (for Admin), and Logout. Must be visible on all screens after login.	 StoreBhai · Search Terms of Service · Privacy Policy · Help & Support · Reports · Help · © StoreBhai 2025
phase 1	8	Bottom Bar	<Bottom Bar> - Copyright (© StoreBhai 2025) - Terms of Service link - Privacy Policy link - Help/Support link - Version info	As a user, I want to see StoreBhai's copyright and legal links so I can know the policies and trust the platform. As an admin, I want quick links to Help/Reports/Support in the bottom bar so I can access them directly.	Copyright must display current year dynamically. Terms link → open Terms of Service page. Privacy link → open Privacy Policy page. Help/Support link → open FAQ/support request (Admin sees "Reports" link here too). Version info (e.g., v1.0.0) must always be shown. Visible consistently across all pages (both app and admin panel).	 StoreBhai · Search Terms of Service · Privacy Policy · Help & Support · Reports · Help · © StoreBhai 2025
Feature: 4	**Payments**					
phase 2	9	Payments[Renter]	<Form> <input = booking id> <input = amount> <select = payment method> (bKash / Nagad) <button = pay> </Form>	As a renter, I want to pay with bKash/Nagad so I can confirm my storage booking easily. As an admin, I want to log all transactions so payments are traceable.	Booking ID must exist. Otherwise → error. Amount must match booking amount. Otherwise → error. Payment method must be selected. On success → show confirmation, update booking status, send receipt via SMS/email. Failed payment → show error and retry option.	 StoreBhai · Rent · List your space · Help Payments Space in Dhaka Dhaka, Bangladesh · Jan 15 - Jan 20 Price breakdown 5 nights BDT 10,000 Service fee BDT 500 Total BDT 10,500 Payment method bKash · Nagad · Rocket · Bank 01XXXXXXXXX Pay Now Funds are held securely until check-in. Recent Payment History Date · Booking · Payment Method · Amount · Status Jan 15, 2024 · Space in Dhaka · bKash · BDT 10,500 · Completed Dec 20, 2023 · Space in Chittagong · Nagad · BDT 8,000 · Completed Nov 5, 2023 · Space in Sylhet · Rocket · BDT 5,000 · Refunded View All Transactions Request Refund
Feature: 10	**Profile**					
phase 1	15	View Profile [Host][Renter]	<Profile> - Name - Phone - Email - DOB - Gender - User Type (Host/Renter) - Edit button </Profile>	As a user, I want to view my profile so I can check my details.	Profile page must show all fields stored in DB. Edit button must redirect to Edit Profile page.	 StoreBhai · Rent · Host · Help Fatima Rahman Host Contact Info Full Name: Fatima Rahman Date of Birth: 1990-05-15 Address / Location: Dhaka, Bangladesh Phone Number: +880 1712345678 Gender: Female Role: Host Edit Profile Security Change Password Two-factor authentication · Enable/Disable Legal Terms of Service · Privacy Policy Logout
phase 1	16	Edit Profile [Host][Renter]	<Form> <input = name> <input = phone> <input = email> <input = dob> <select = gender> <button = confirm> </Form>	As a user, I want to edit my profile so I can keep my details up to date.	Fields must show current values pre-filled. Phone/email format validation. On clicking confirm → details updated in DB. Show success message "Profile updated."	 StoreBhai · Rent · Host · Help Edit Profile Full Name Phone Number Email Address Date of Birth Gender Address / Location Cancel · Save Changes

phase 1	17	Change Profile Photo [Host][Renter]	<Form> <input type=file> <button = upload> </Form>	As a user, I want to change my photo so others can recognize my account.	Only image formats allowed (.jpg, .png). On success → new photo displayed in profile. Confirm button returns to profile view.	 StoreBhai Change Profile Picture Upload a new photo to update your profile. Change Photo Choose File Drag & Drop JPG, PNG Take Photo Crop & Rotate Rotate 50 Cancel Confirm
phase 1	18	Change Password [Host][Renter]	<Form> <input = current password> <input = new password> <input = re-password> <button = confirm> </Form>	As a user, I want to change my password so I can keep my account secure.	Current password must match. Otherwise → "Incorrect current password." New password must meet policy (≥ 8 chars). Re-password must match. Otherwise → "Passwords do not match." On success → password updated in DB, user redirected to profile page.	 StoreBhai Change Password Update your password to keep your account secure. Current Password Enter yourcurrentpassword New Password Enter yournewpassword Password Strength Medium Confirm New Password Confirm yournewpassword Update Password Back to Profile Need help? Contact Support

PROJECT ESTIMATION AND SCHEDULING

Effort and Cost Estimation Scope Definition

Scope Definition

The Vehicle Rental System will allow customers to rent vehicles online, manage bookings, track rental history, generate bills, and allow admins to manage the fleet, users, and financials.

LOC Estimation

Estimated Size = 12000 LOC = 12 KLO
Productivity rate (assumption) = 2000 LOC/person-month

Effort using COCOMO Method

Formula used: E = ((LOC × B^0.333) / P)^3 × (1/t^4) Where: LOC = 12,000 B = 2.5 (complexity constant) P = 2000 (productivity) t = 3 months (duration). E ≈ 5 person-months

Development Time (TDEV) & Staffing

TDEV = 2.5 × (E^0.38) ≈4. 6 months - With 10 people → ~1.5 months - With 5 people → ~3 months

Cost Estimation

If average cost per person-month = $2,000: Total Cost = 5 × 2000 = $10,000

Project Scheduling

Task Breakdown (WBS style)

1. Analysis & Design (40%) → Requirement gathering, UML diagrams, database schema.
2. Coding & Implementation (20%) → Backend, frontend, integration.
3. Testing & Deployment (40%) → Unit testing, integration testing, UAT, deployment.

Effort Allocation

Total Effort = 5 person-months
- Analysis/Design = 40% → 2 PM
- Coding = 20% → 1 PM
- Testing/Deployment = 40% → 2 PM

Schedule Representation

Tasks are connected as: Design → Coding → Testing → Deployment.
Effort distribution follows the 40–20–40 guideline.

Earned Value Analysis (EVA)
Assumptions for mid-project (50% timeline):
- BAC (Budget at Completion) = $15000
- BCWS (Planned Value) = $12000
- BCWP (Earned Value) = $10000
- ACWP (Actual Cost) =$7000

EVA Metrics

SPI = BCWP / BCWS = 10M / 12M = 0.83

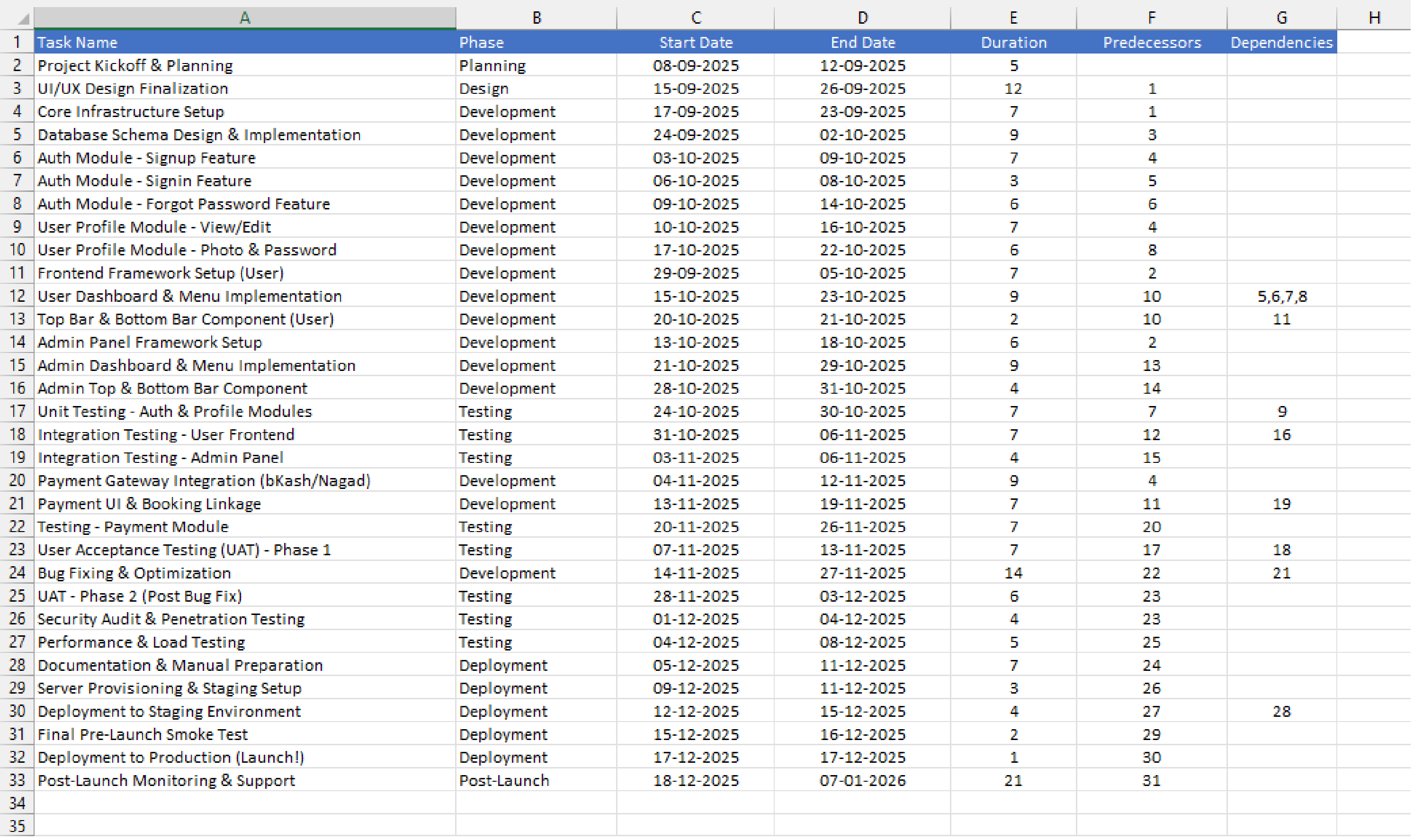
SV = BCWP – BCWS = 12M – 15M = –3M (behind schedule)

CPI = BCWP / ACWP = 12M / 14M = 0.857

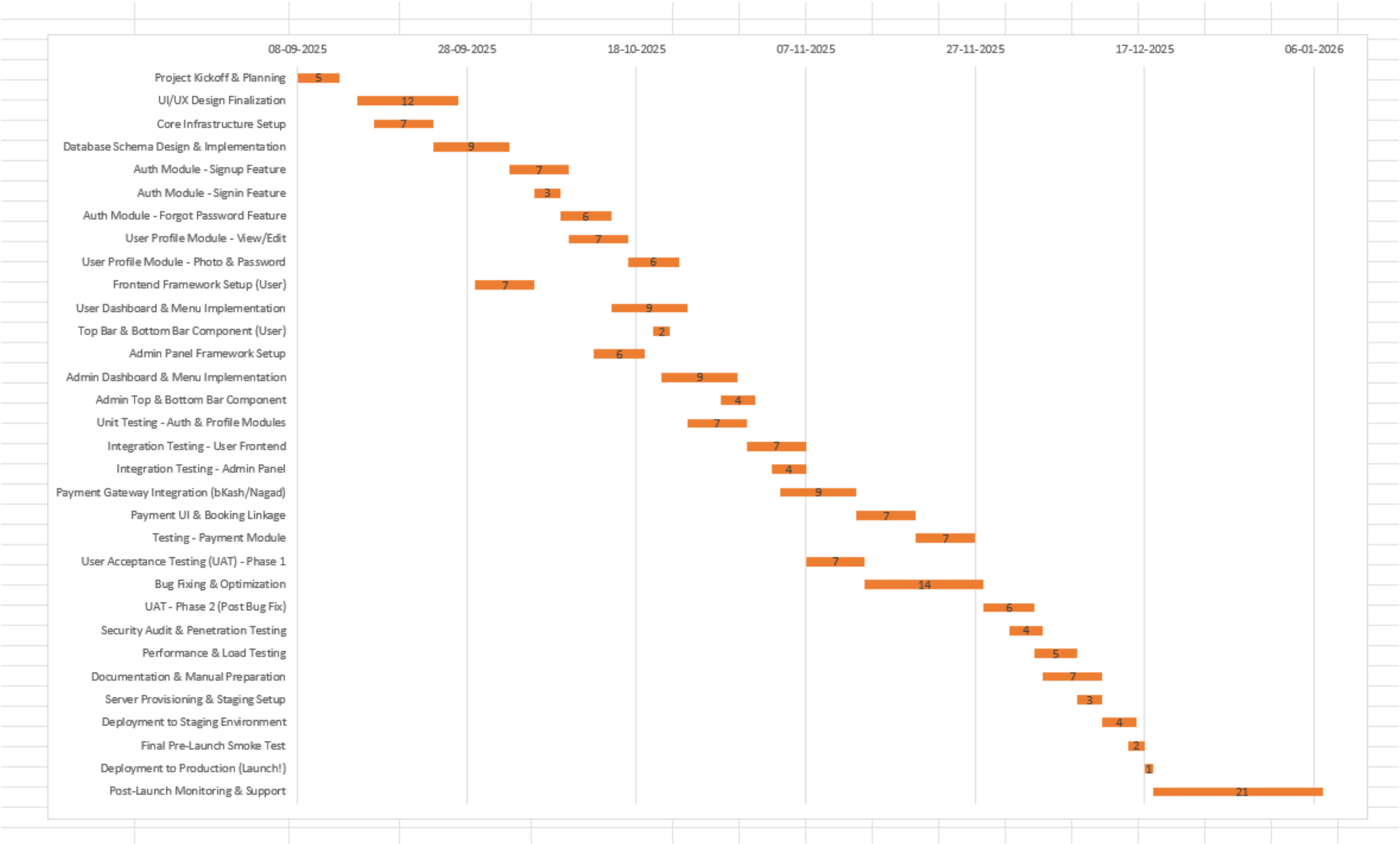
CV = BCWP – ACWP = 12M – 7M = 5M (under budget)

% Schedule Completion = BCWS / BAC = 12M / 15M≈ 80%

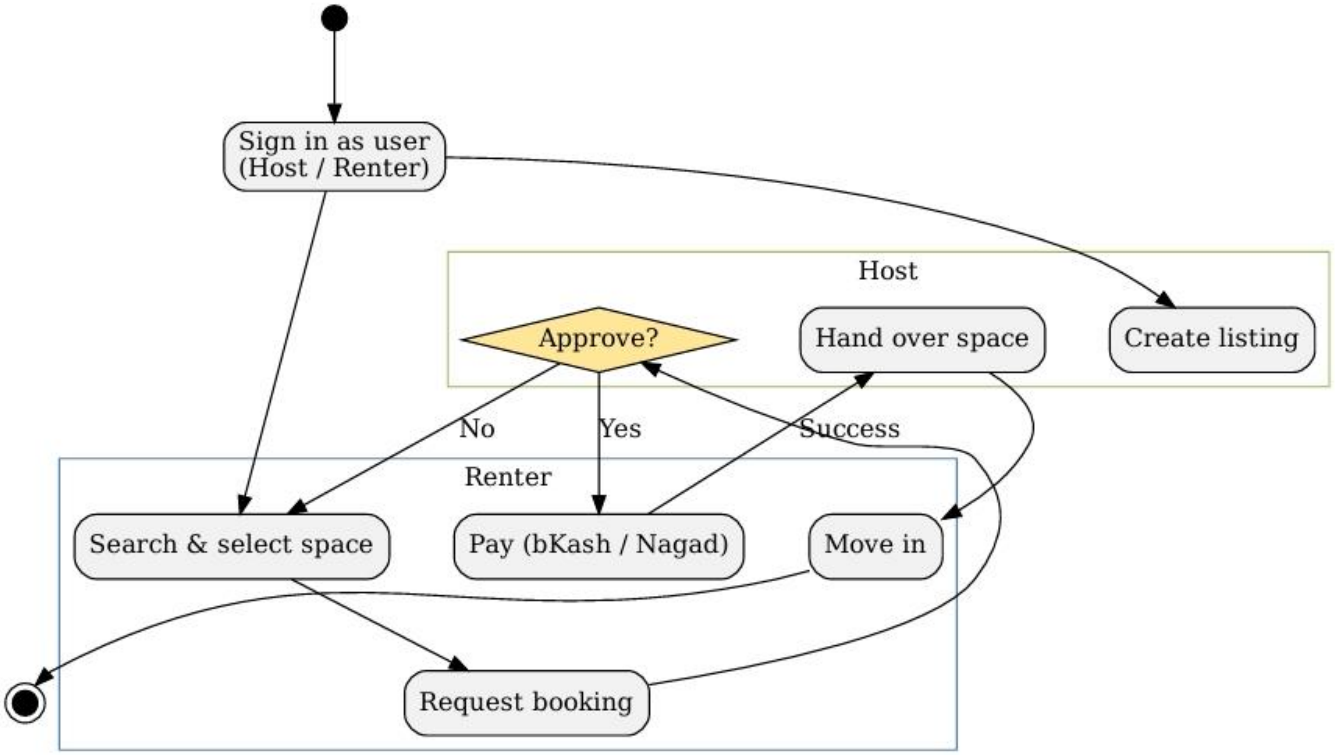
% Complete = BCWP / BAC = 10M / 15M ≈ 66.6%



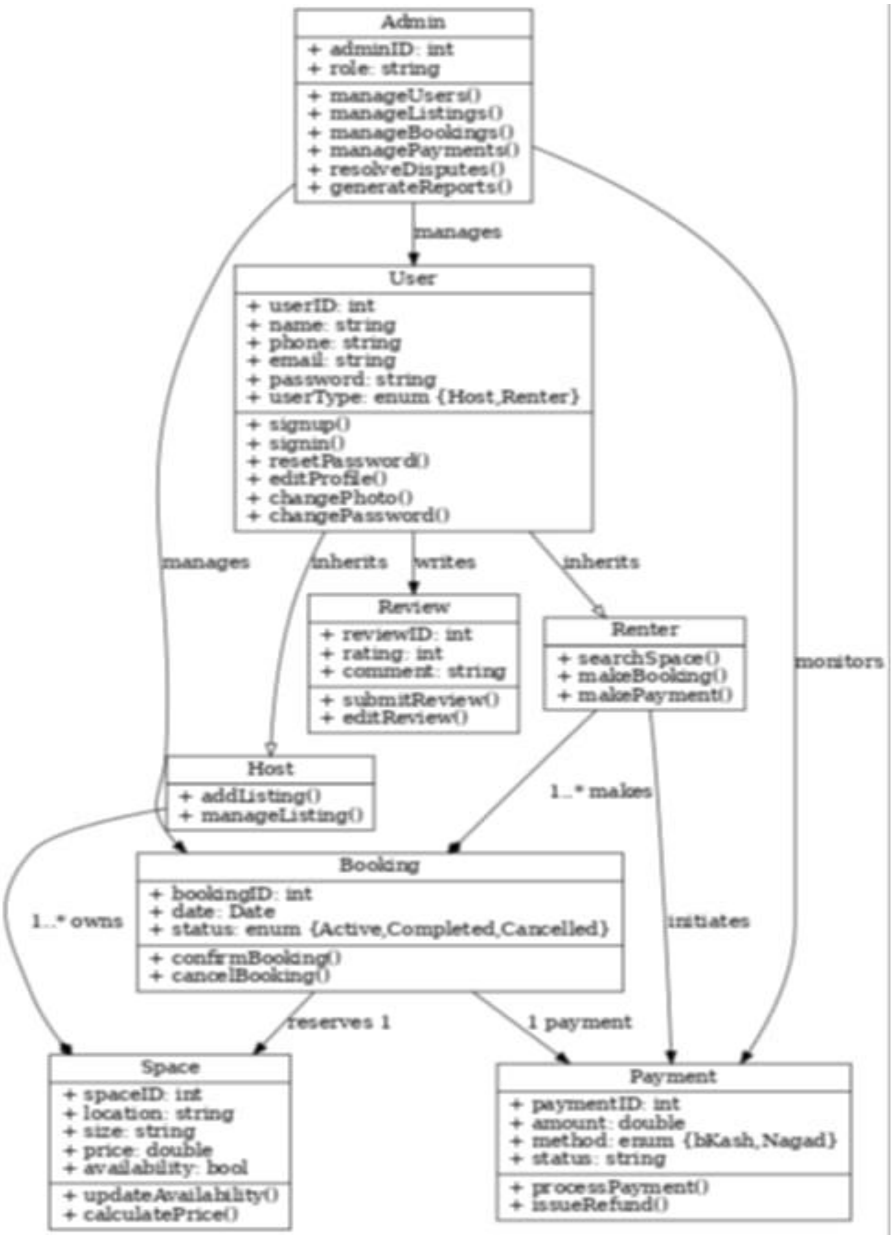
	A	B	C	D	E	F	G	H
1	Task Name	Phase	Start Date	End Date	Duration	Predecessors	Dependencies	
2	Project Kickoff & Planning	Planning	08-09-2025	12-09-2025	5			
3	UI/UX Design Finalization	Design	15-09-2025	26-09-2025	12	1		
4	Core Infrastructure Setup	Development	17-09-2025	23-09-2025	7	1		
5	Database Schema Design & Implementation	Development	24-09-2025	02-10-2025	9	3		
6	Auth Module - Signup Feature	Development	03-10-2025	09-10-2025	7	4		
7	Auth Module - Signin Feature	Development	06-10-2025	08-10-2025	3	5		
8	Auth Module - Forgot Password Feature	Development	09-10-2025	14-10-2025	6	6		
9	User Profile Module - View/Edit	Development	10-10-2025	16-10-2025	7	4		
10	User Profile Module - Photo & Password	Development	17-10-2025	22-10-2025	6	8		
11	Frontend Framework Setup (User)	Development	29-09-2025	05-10-2025	7	2		
12	User Dashboard & Menu Implementation	Development	15-10-2025	23-10-2025	9	10	5,6,7,8	
13	Top Bar & Bottom Bar Component (User)	Development	20-10-2025	21-10-2025	2	10	11	
14	Admin Panel Framework Setup	Development	13-10-2025	18-10-2025	6	2		
15	Admin Dashboard & Menu Implementation	Development	21-10-2025	29-10-2025	9	13		
16	Admin Top & Bottom Bar Component	Development	28-10-2025	31-10-2025	4	14		
17	Unit Testing - Auth & Profile Modules	Testing	24-10-2025	30-10-2025	7	7	9	
18	Integration Testing - User Frontend	Testing	31-10-2025	06-11-2025	7	12	16	
19	Integration Testing - Admin Panel	Testing	03-11-2025	06-11-2025	4	15		
20	Payment Gateway Integration (bKash/Nagad)	Development	04-11-2025	12-11-2025	9	4		
21	Payment UI & Booking Linkage	Development	13-11-2025	19-11-2025	7	11	19	
22	Testing - Payment Module	Testing	20-11-2025	26-11-2025	7	20		
23	User Acceptance Testing (UAT) - Phase 1	Testing	07-11-2025	13-11-2025	7	17	18	
24	Bug Fixing & Optimization	Development	14-11-2025	27-11-2025	14	22	21	
25	UAT - Phase 2 (Post Bug Fix)	Testing	28-11-2025	03-12-2025	6	23		
26	Security Audit & Penetration Testing	Testing	01-12-2025	04-12-2025	4	23		
27	Performance & Load Testing	Testing	04-12-2025	08-12-2025	5	25		
28	Documentation & Manual Preparation	Deployment	05-12-2025	11-12-2025	7	24		
29	Server Provisioning & Staging Setup	Deployment	09-12-2025	11-12-2025	3	26		
30	Deployment to Staging Environment	Deployment	12-12-2025	15-12-2025	4	27	28	
31	Final Pre-Launch Smoke Test	Deployment	15-12-2025	16-12-2025	2	29		
32	Deployment to Production (Launch!)	Deployment	17-12-2025	17-12-2025	1	30		
33	Post-Launch Monitoring & Support	Post-Launch	18-12-2025	07-01-2026	21	31		
34								
35								



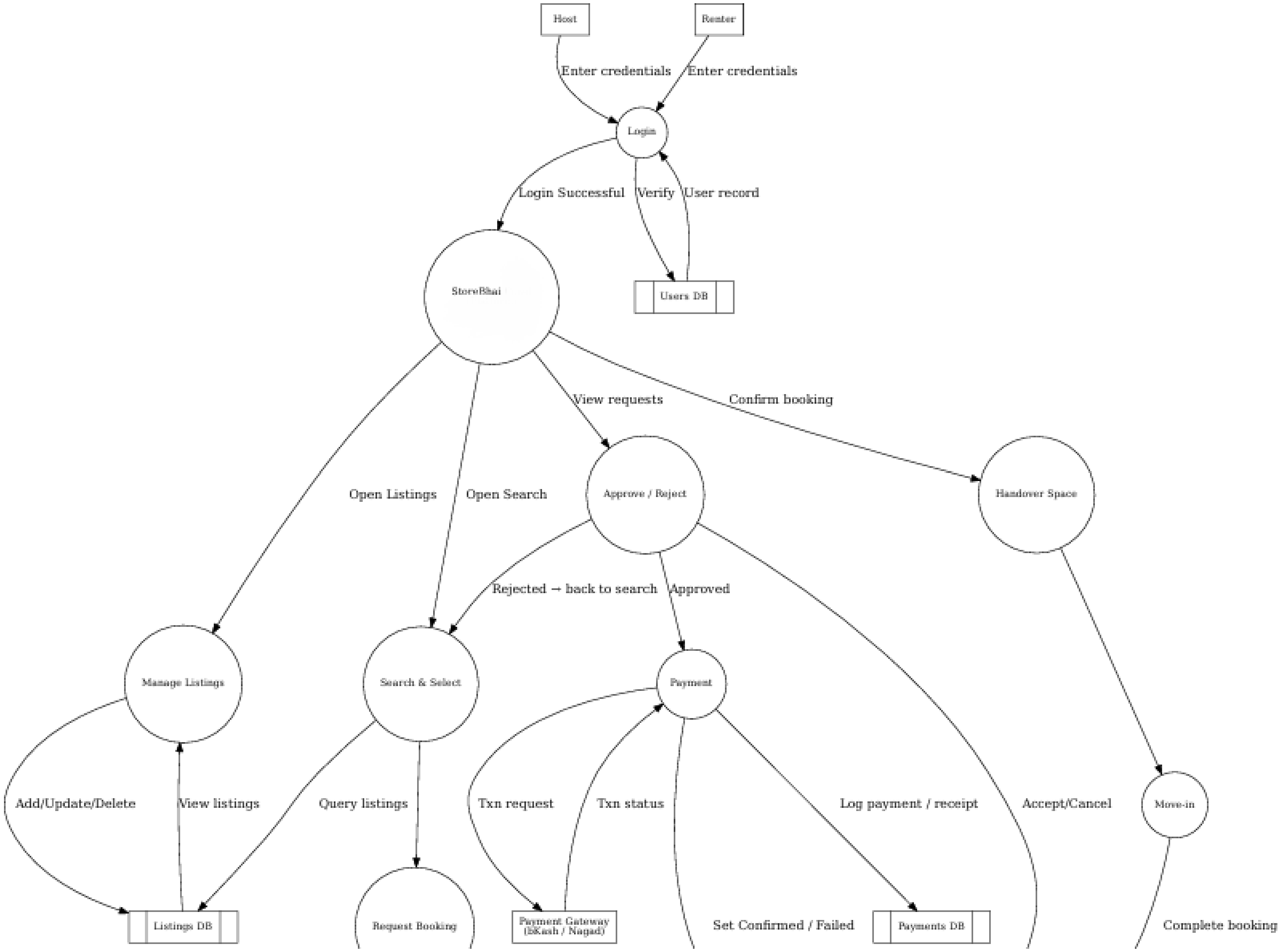
Activity Diagram



Class Diagram

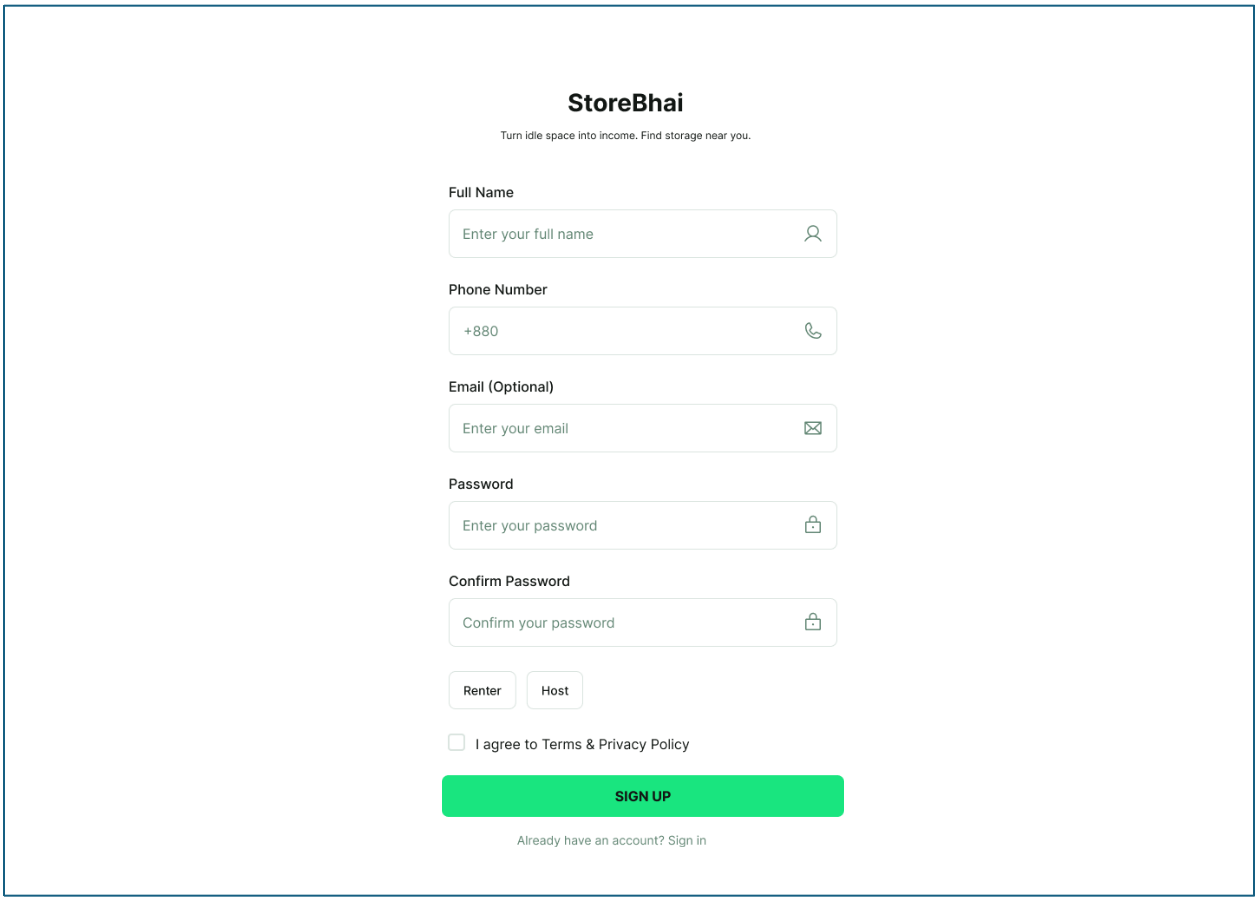


Data Flow Diagram


Host
Renter
Enter credentials
Enter credentials
Login
Login Successful
Verify
User record
StoreBhai
Users DB
View requests
Confirm booking
Open Listings
Open Search
Approve / Reject
Handover Space
Rejected → back to search
Approved
Manage Listings
Search & Select
Payment
Add/Update/Delete
View listings
Query listings
Txn request
Txn status
Log payment / receipt
Accept/Cancel
Move-in
Listings DB
Request Booking
Payment Gateway
(bKash / Nagad)
Set Confirmed / Failed
Payments DB
Complete booking

StoreBhai

SignUp:



StoreBhai

Turn idle space into income. Find storage near you.

Full Name

Enter your full name

Phone Number

+880

Email (Optional)

Enter your email

Password

Enter your password

Confirm Password

Confirm your password

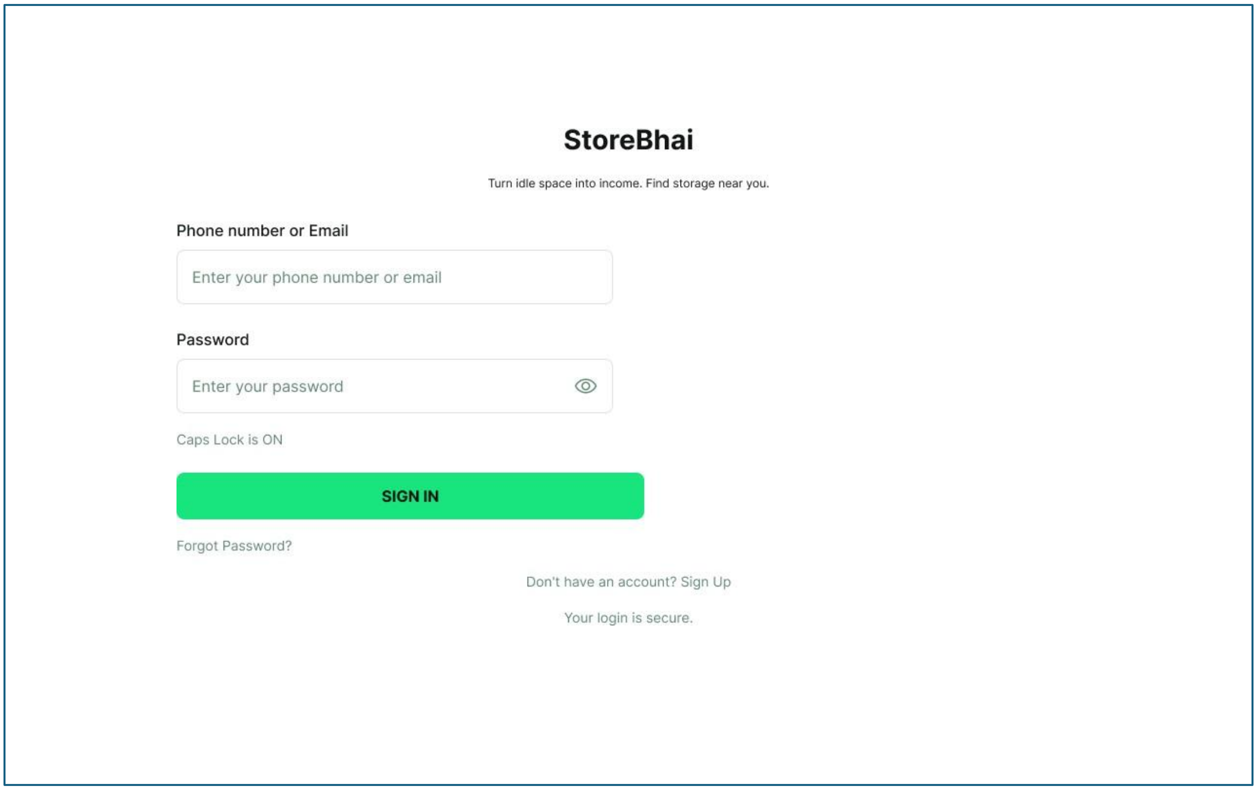
Renter Host

□ I agree to Terms & Privacy Policy

SIGN UP

Already have an account? Sign in

Sign In :



StoreBhai

Turn idle space into income. Find storage near you.

Phone number or Email

Enter your phone number or email

Password

Enter your password

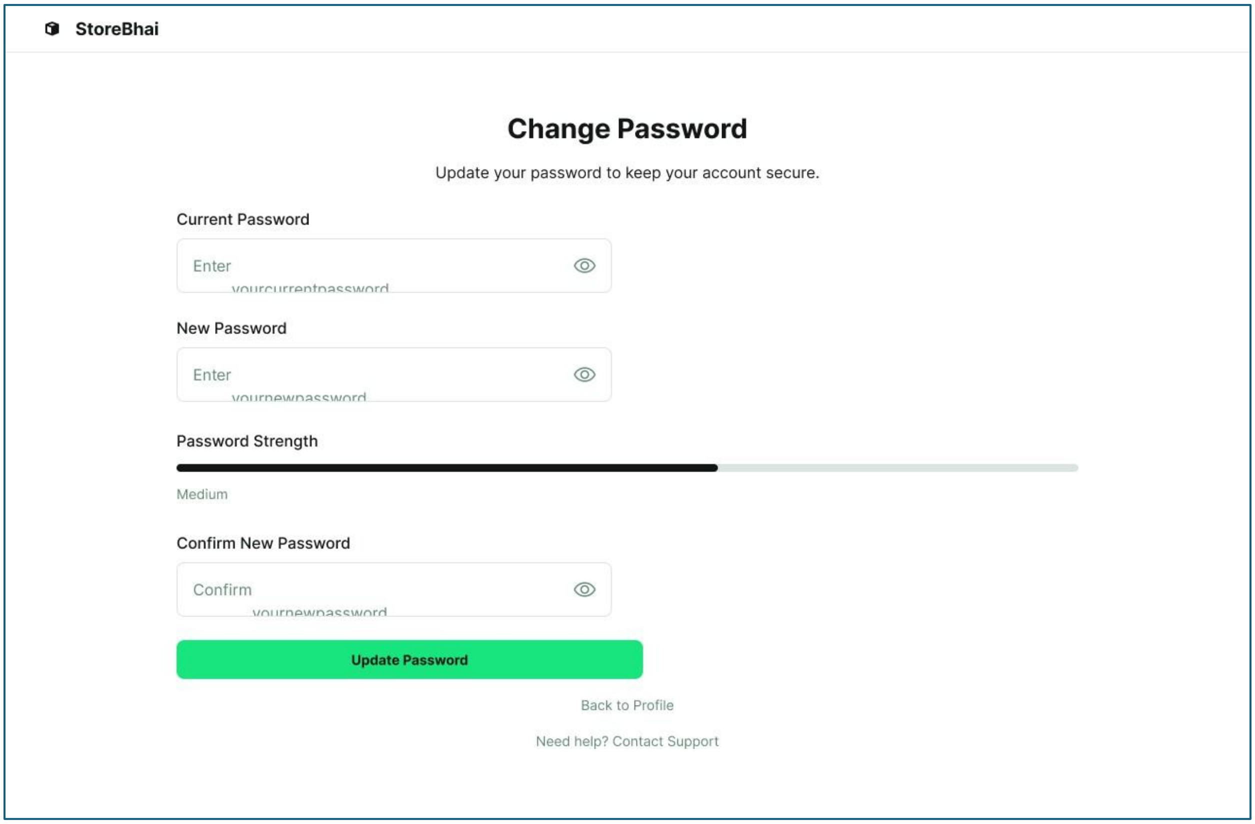
Caps Lock is ON

Forgot Password?

Don't have an account? Sign Up

Your login is secure.

Change Password :



StoreBhai

Change Password

Update your password to keep your account secure.

Current Password

Enter yourcurrentpassword

New Password

Enter yournewpassword

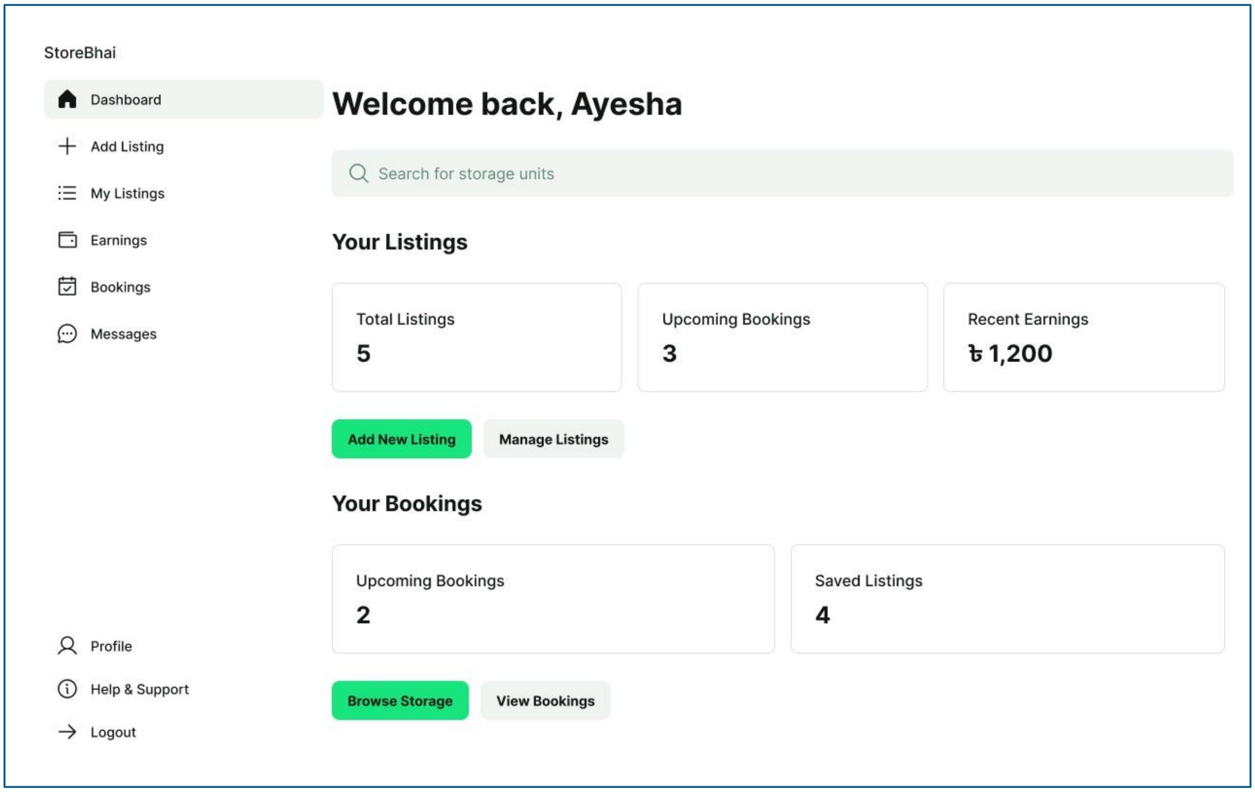
Password Strength

Medium

Confirm New Password

Confirm yournewpassword

Menu [Host] [Renter]



Welcome back, Ayesha

Your Listings

Total Listings
5

Upcoming Bookings
3

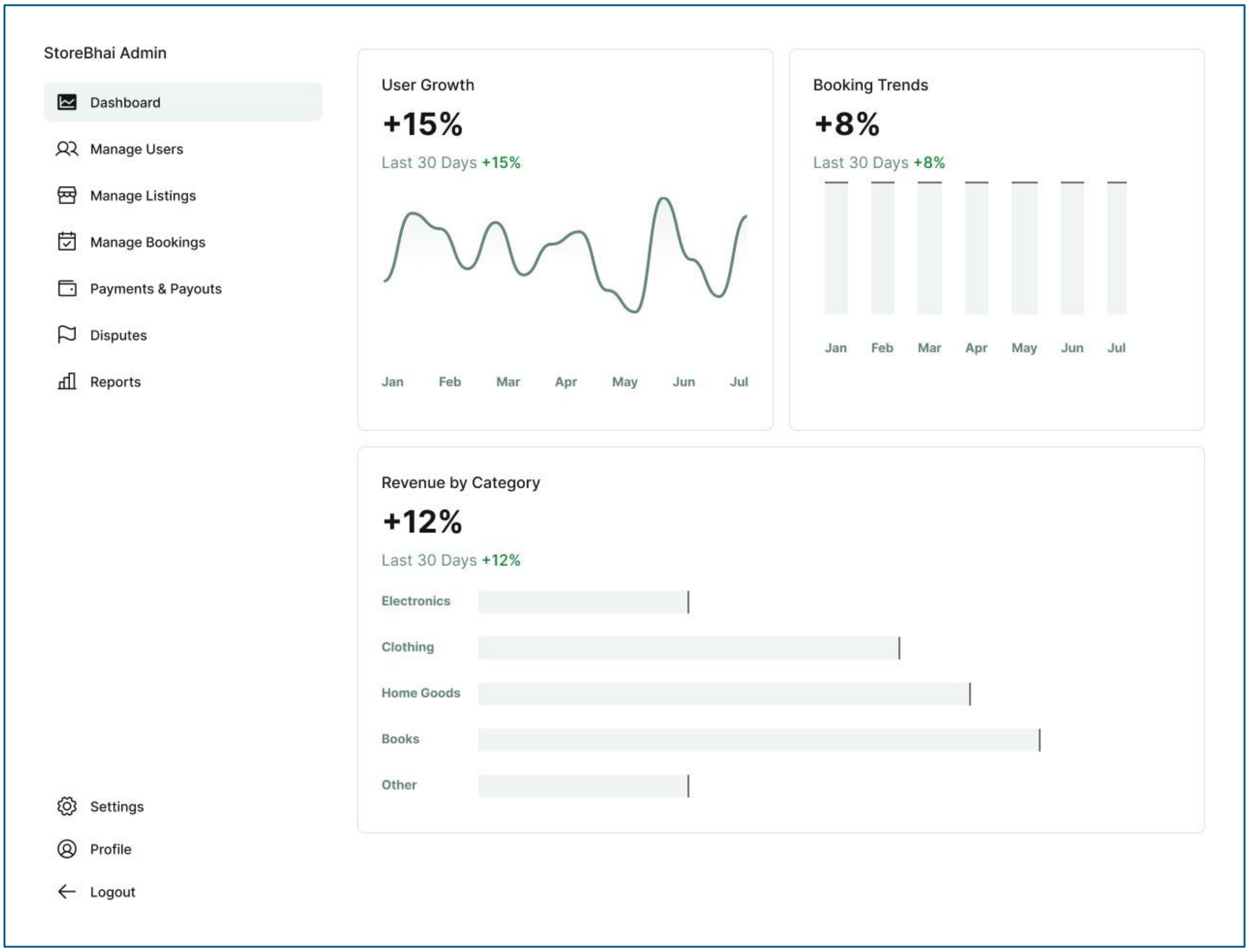
Recent Earnings
৳ 1,200

Your Bookings

Upcoming Bookings
2

Saved Listings
4

Menu [Admin]



User Growth

+15%

Last 30 Days +15%

Booking Trends

+8%

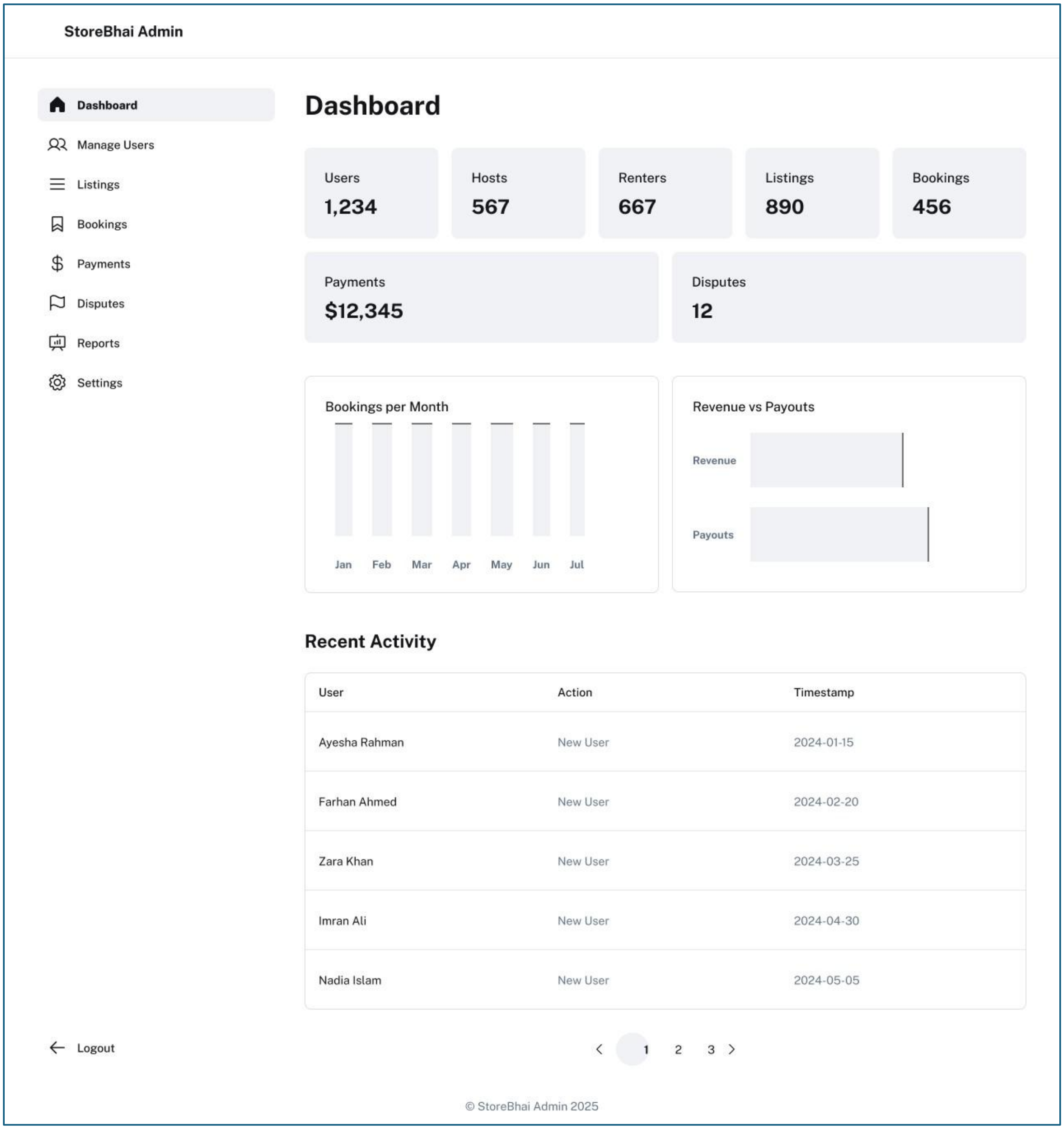
Last 30 Days +8%

Revenue by Category

+12%

Last 30 Days +12%

Dashboard [Admin]



StoreBhai Admin

Dashboard

Users 1,234	Hosts 567	Renters 667	Listings 890	Bookings 456

Payments $12,345	Disputes 12

Bookings per Month

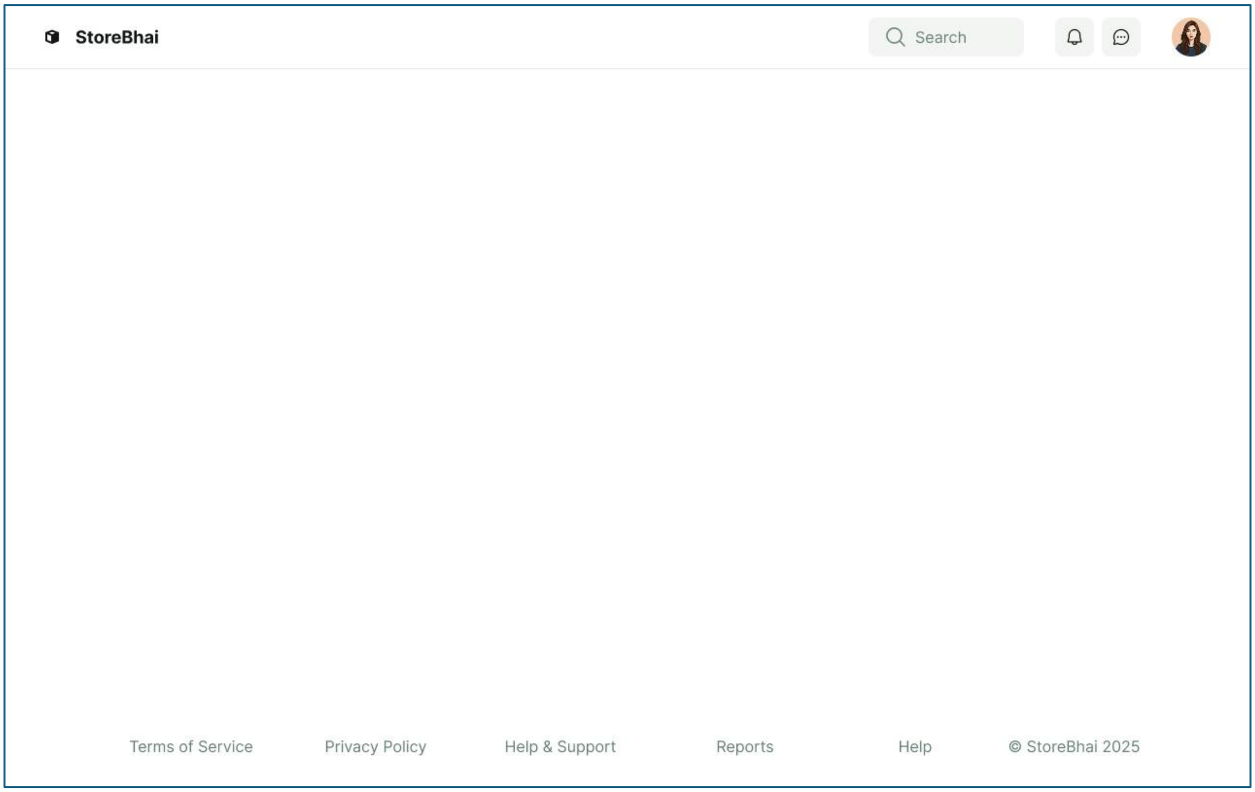
Revenue vs Payouts

Recent Activity

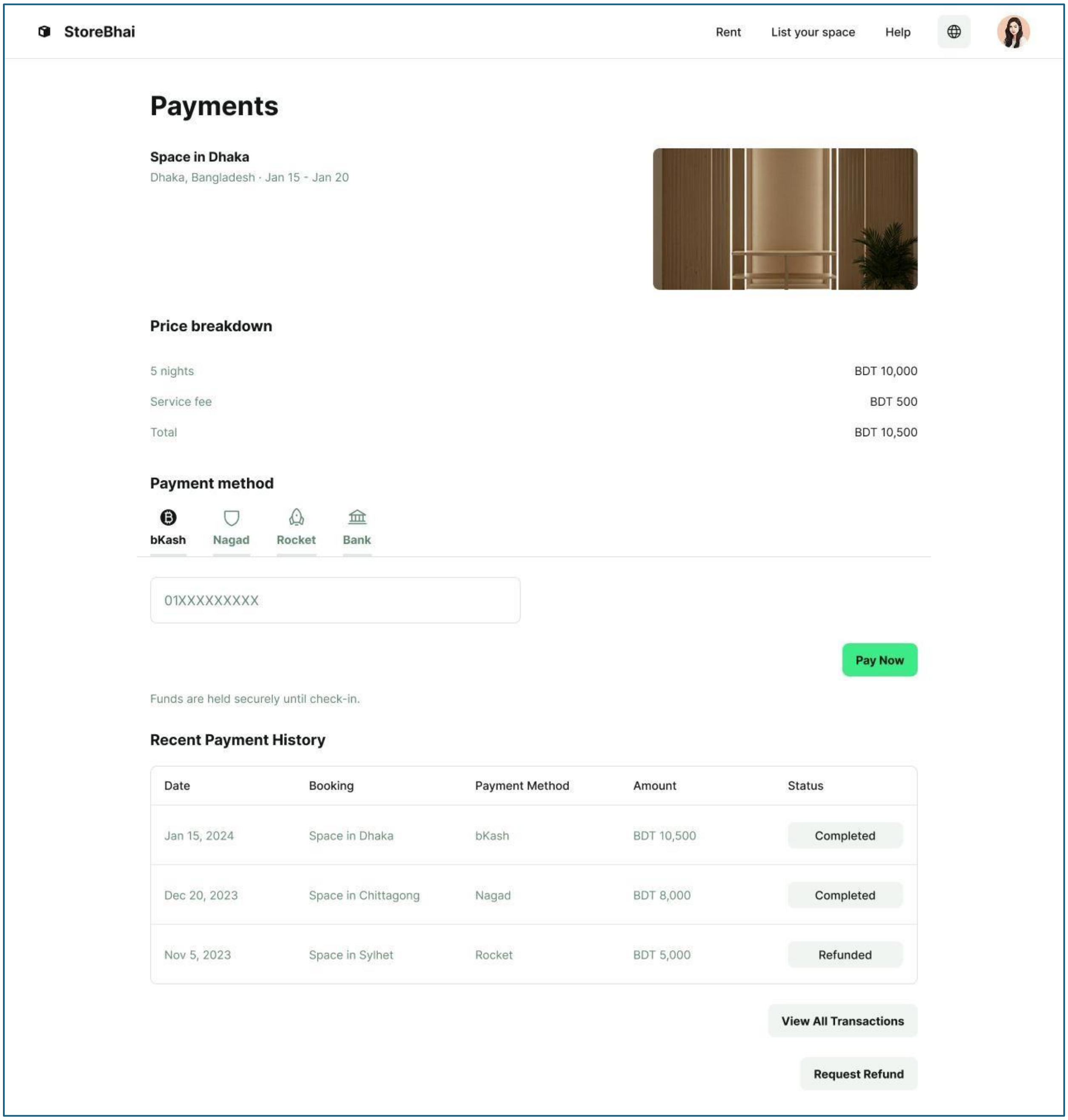
User	Action	Timestamp
Ayesha Rahman	New User	2024-01-15
Farhan Ahmed	New User	2024-02-20
Zara Khan	New User	2024-03-25
Imran Ali	New User	2024-04-30
Nadia Islam	New User	2024-05-05

© StoreBhai Admin 2025

Top Bar + Bottom Bar:



Payments:



Payments

Space in Dhaka

Dhaka, Bangladesh · Jan 15 - Jan 20

Price breakdown

5 nights	BDT 10,000
Service fee	BDT 500
Total	BDT 10,500

Payment method

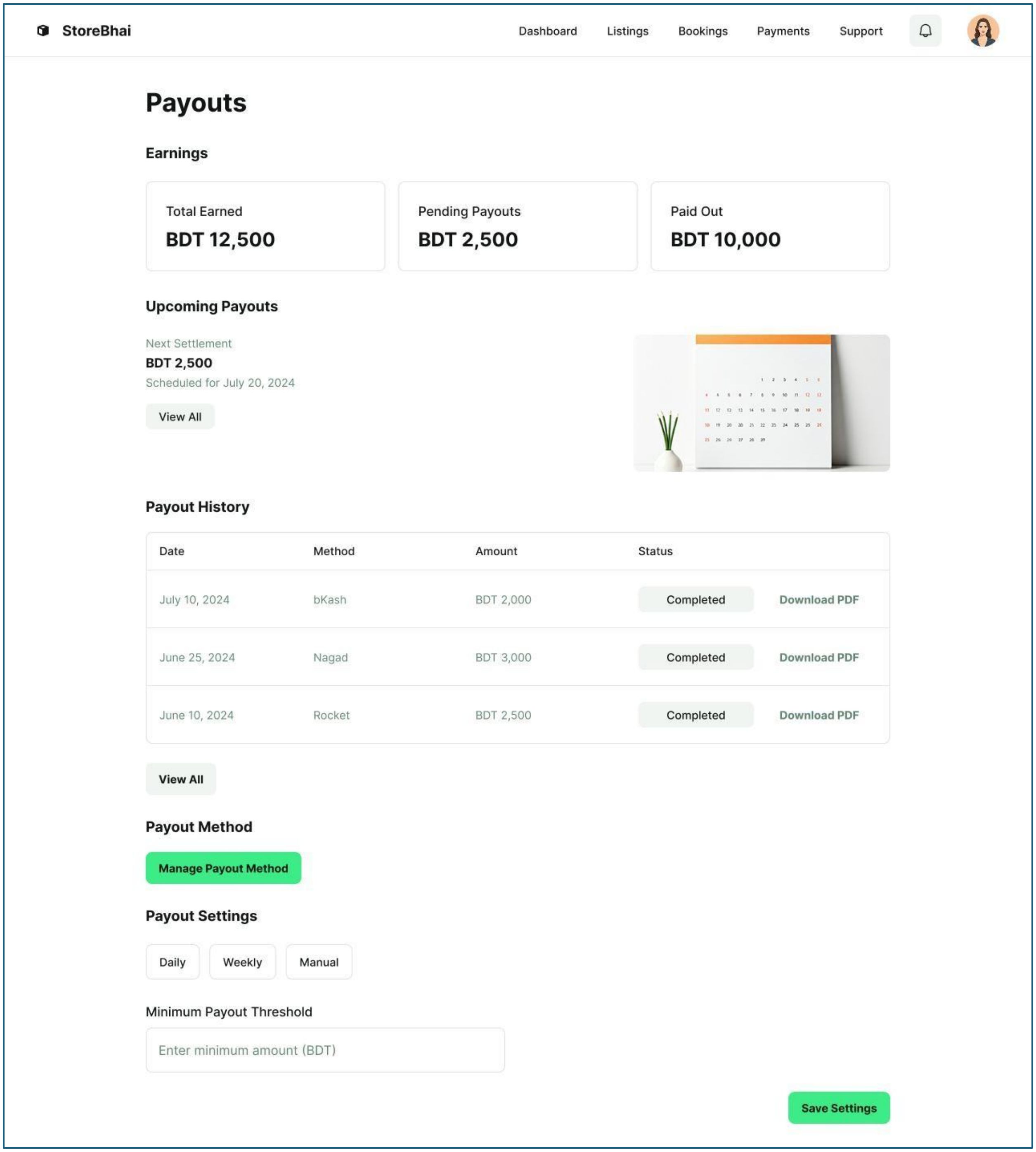
01XXXXXXXXX

Funds are held securely until check-in.

Recent Payment History

Date	Booking	Payment Method	Amount	Status
Jan 15, 2024	Space in Dhaka	bKash	BDT 10,500	Completed
Dec 20, 2023	Space in Chittagong	Nagad	BDT 8,000	Completed
Nov 5, 2023	Space in Sylhet	Rocket	BDT 5,000	Refunded

Payouts:



Payouts

Earnings

Total Earned
BDT 12,500

Pending Payouts
BDT 2,500

Paid Out
BDT 10,000

Upcoming Payouts

Next Settlement
BDT 2,500
Scheduled for July 20, 2024

Payout History

Date	Method	Amount	Status	
July 10, 2024	bKash	BDT 2,000	Completed	Download PDF
June 25, 2024	Nagad	BDT 3,000	Completed	Download PDF
June 10, 2024	Rocket	BDT 2,500	Completed	Download PDF

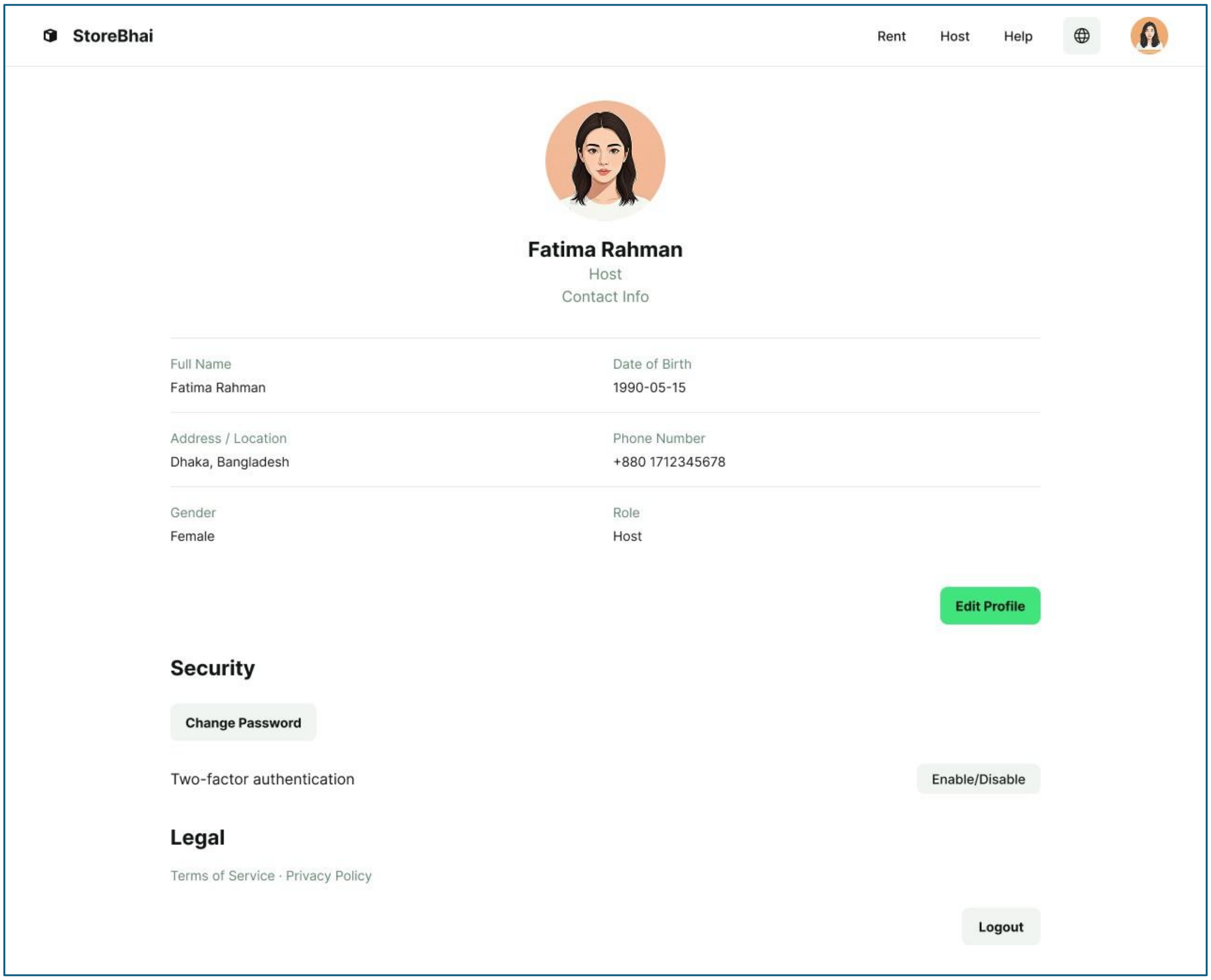
Payout Method

Payout Settings

Minimum Payout Threshold

Enter minimum amount (BDT)

Profile:



StoreBhai

Fatima Rahman

Host

Contact Info

Full Name Fatima Rahman	Date of Birth 1990-05-15
Address / Location Dhaka, Bangladesh	Phone Number +880 1712345678
Gender Female	Role Host

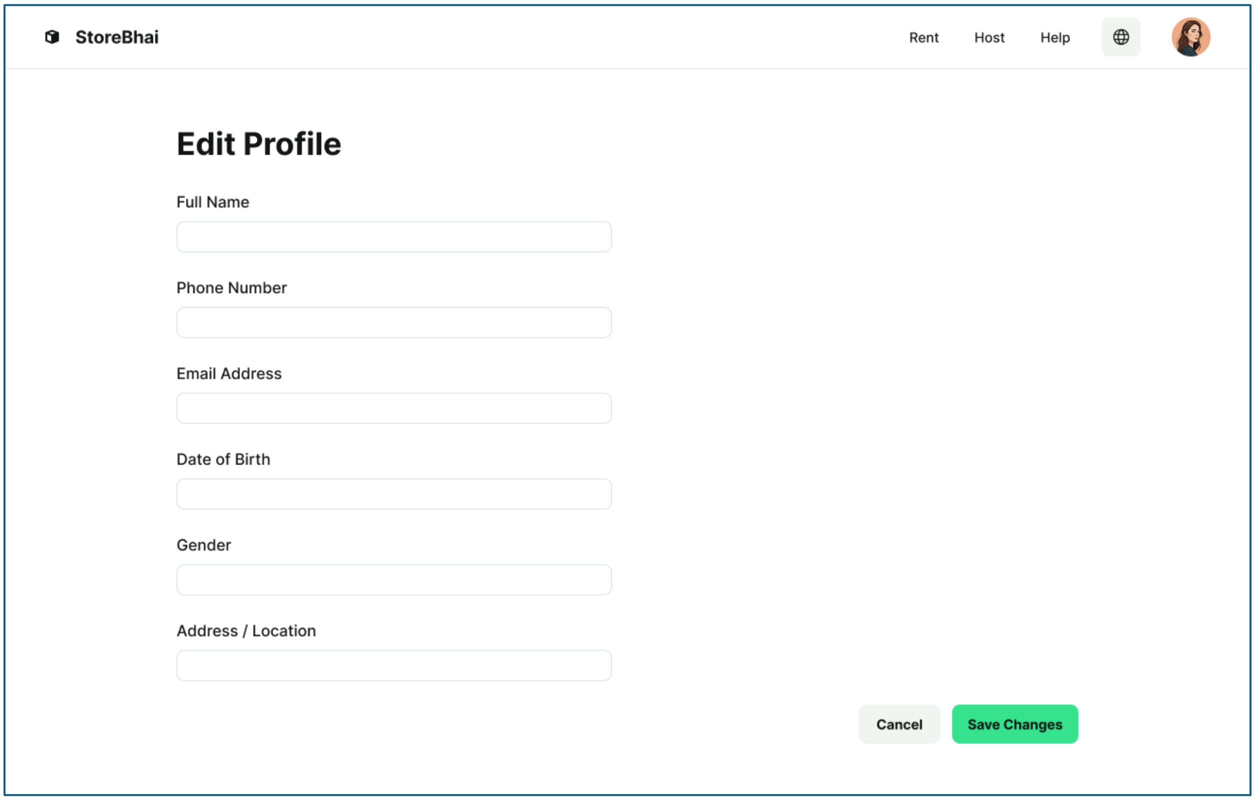
Security

Two-factor authentication

Legal

Terms of Service · Privacy Policy

Edit Profile:



Edit Profile

Full Name

Phone Number

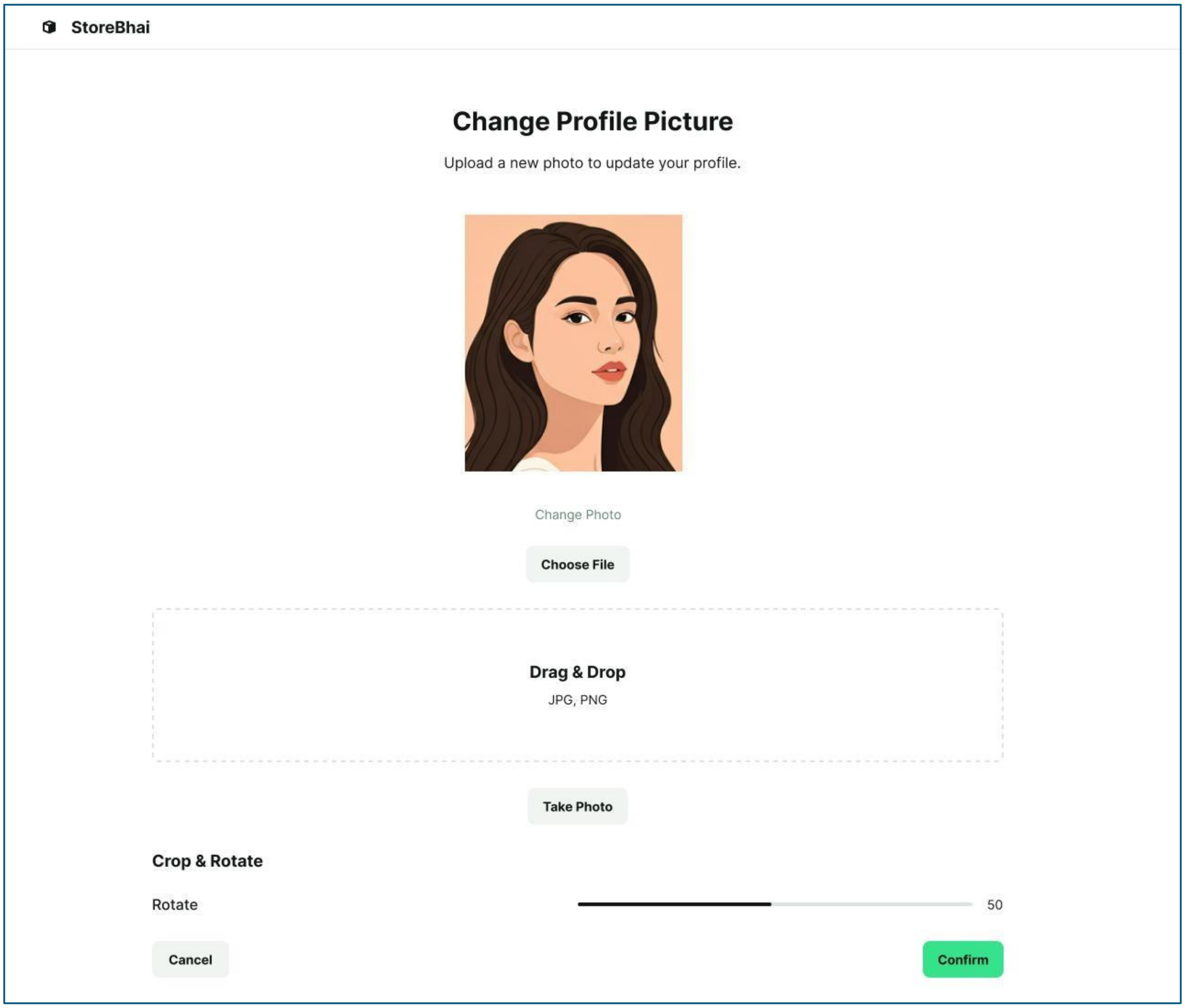
Email Address

Date of Birth

Gender

Address / Location

Change Profile Picture:



StoreBhai

Change Profile Picture

Upload a new photo to update your profile.

Change Photo

Drag & Drop

JPG, PNG

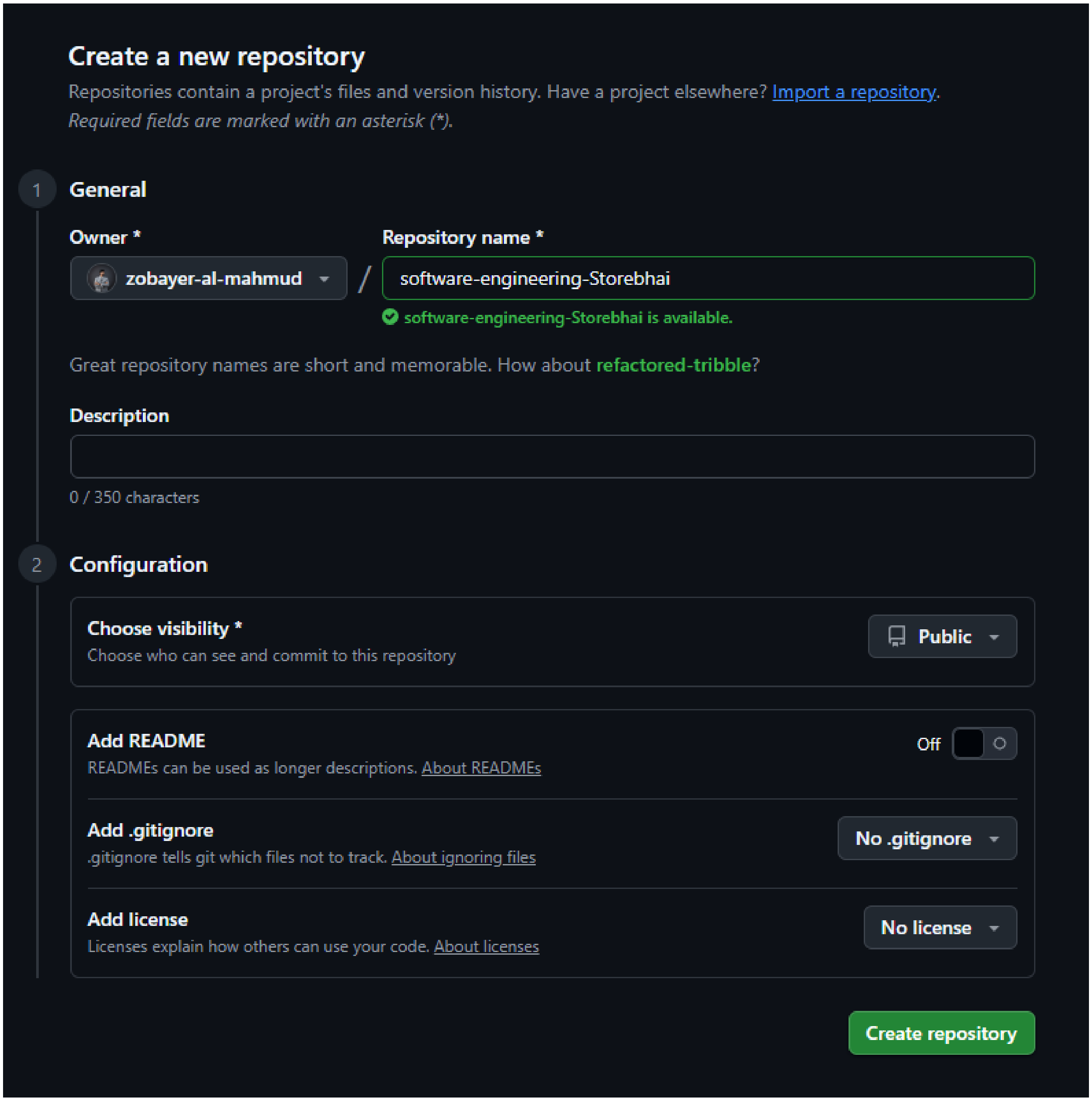
Crop & Rotate

Rotate 50

Figma Link : https://www.figma.com/design/ipHsroWGrCrvxN052IU4br/StoreBhai?node-id=0-1Ct=nQI9EMiVK1jRkLba-1

3. GIT WORKFLOW

https://github.com/zobayer-al-mahmud/software-engineering-Storebhai



Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

1 General

Owner *: zobayer-al-mahmud

Repository name *: software-engineering-Storebhai

software-engineering-Storebhai is available.

Great repository names are short and memorable. How about refactored-tribble?

Description

0 / 350 characters

2 Configuration

Choose visibility *: Public

Choose who can see and commit to this repository

Add README: Off

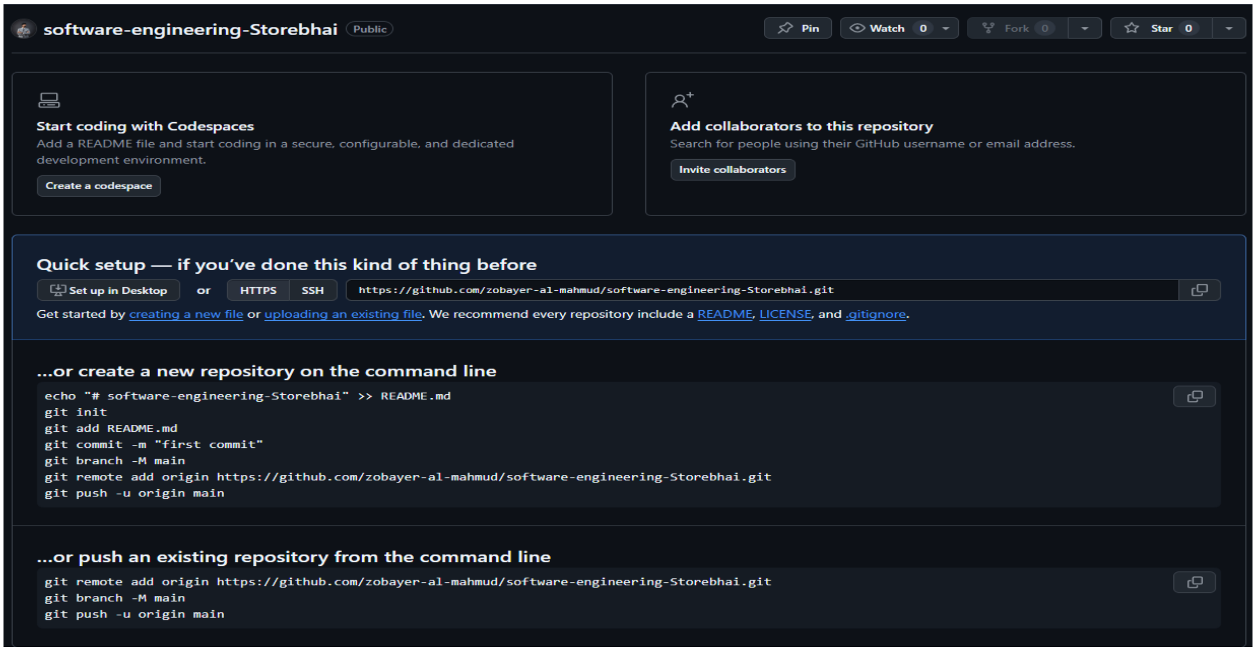
READMEs can be used as longer descriptions. About READMEs

Add .gitignore: No .gitignore

.gitignore tells git which files not to track. About ignoring files

Add license: No license

Licenses explain how others can use your code. About licenses



Start coding with Codespaces

Add a README file and start coding in a secure, configurable, and dedicated development environment.

Add collaborators to this repository

Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

```
https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
```

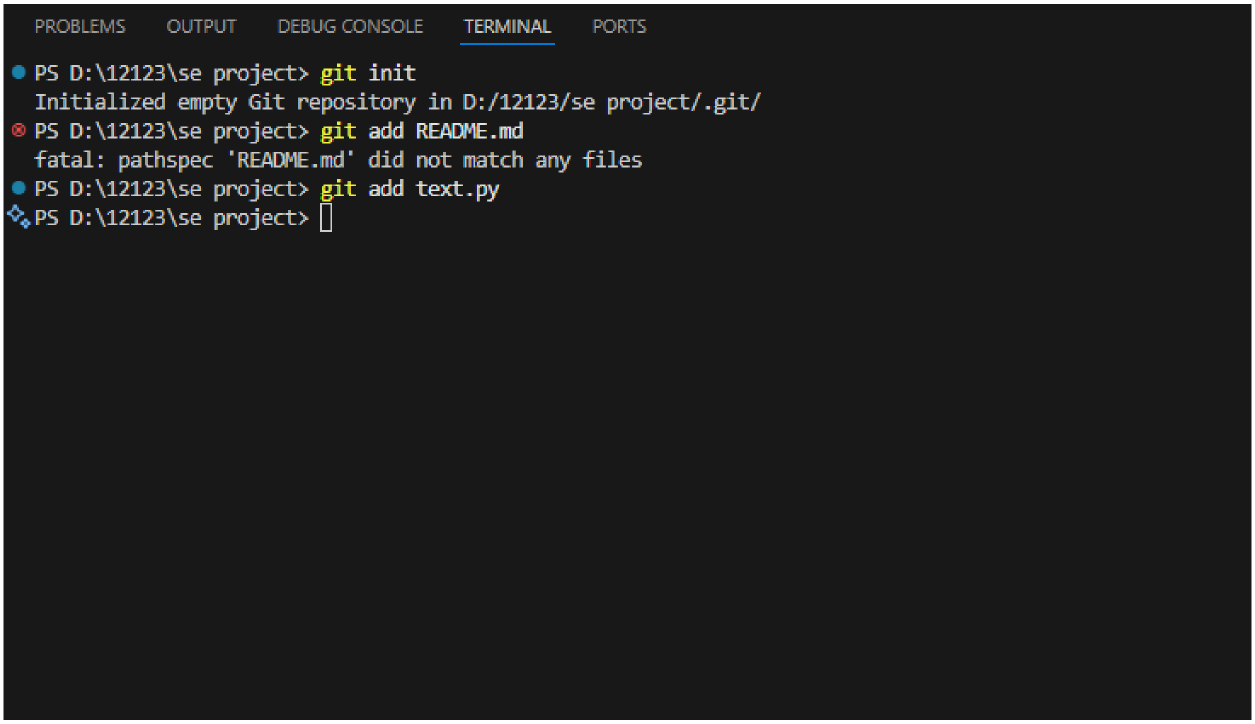
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

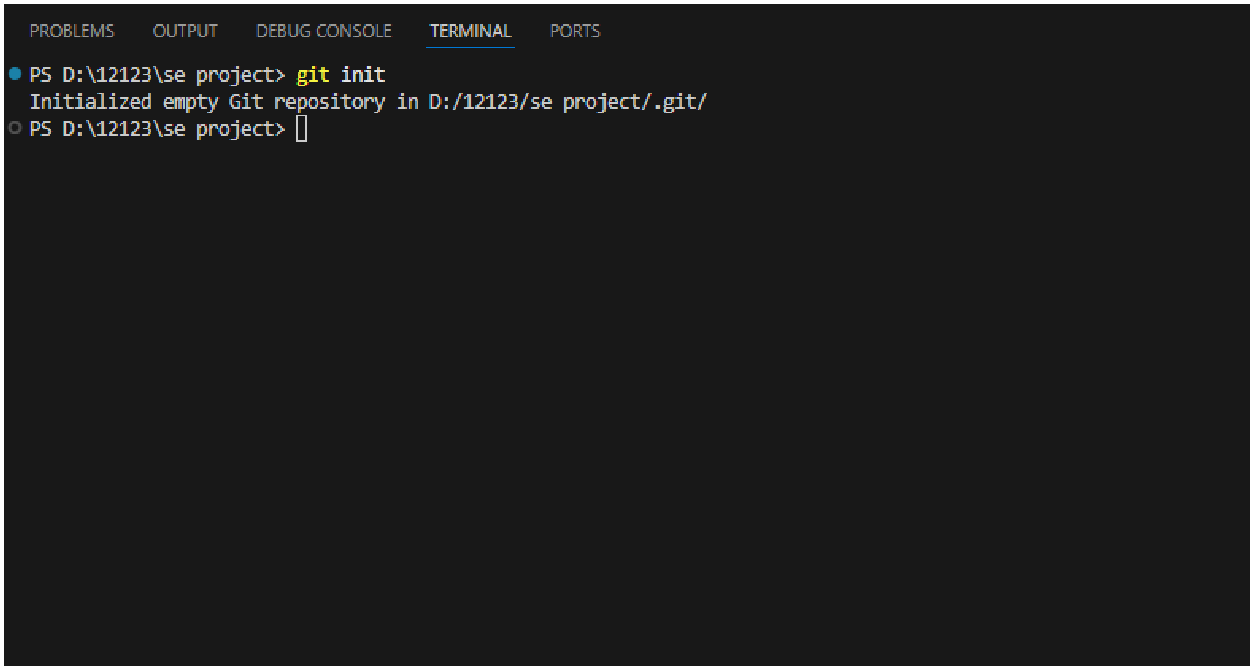
```
echo "# software-engineering-Storebhai" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
git push -u origin main
```

...or push an existing repository from the command line

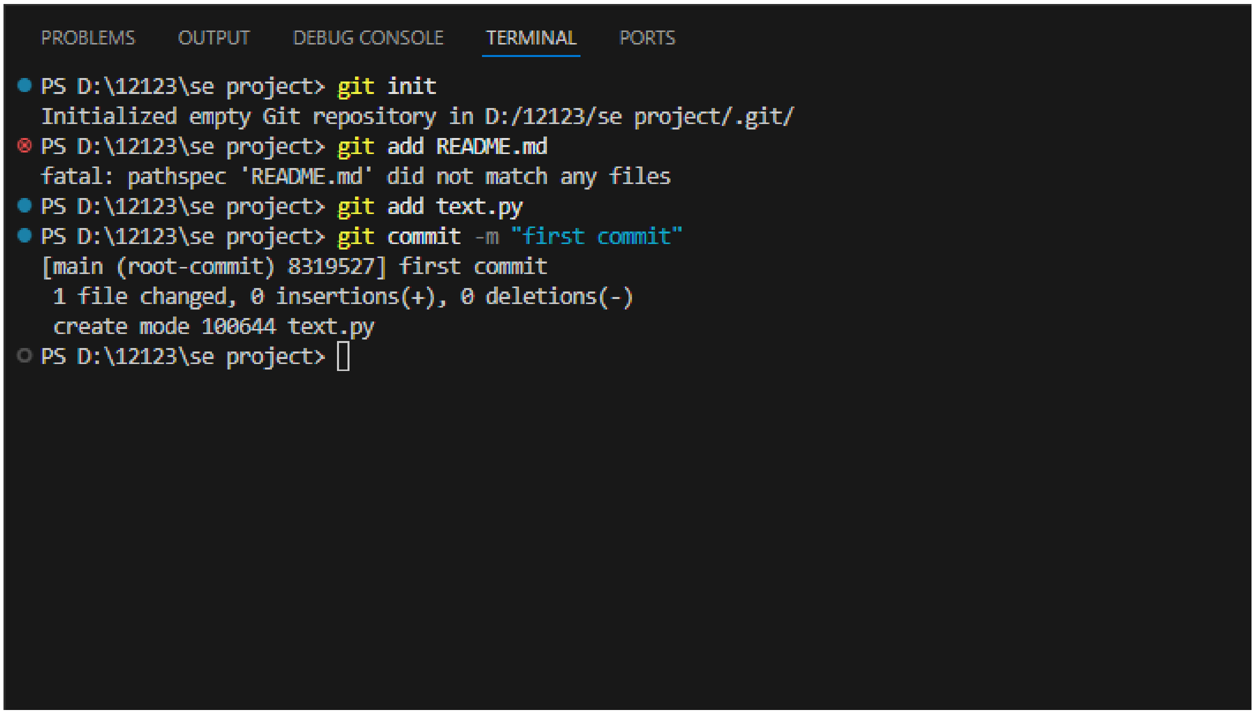
```
git remote add origin https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
git branch -M main
git push -u origin main
```



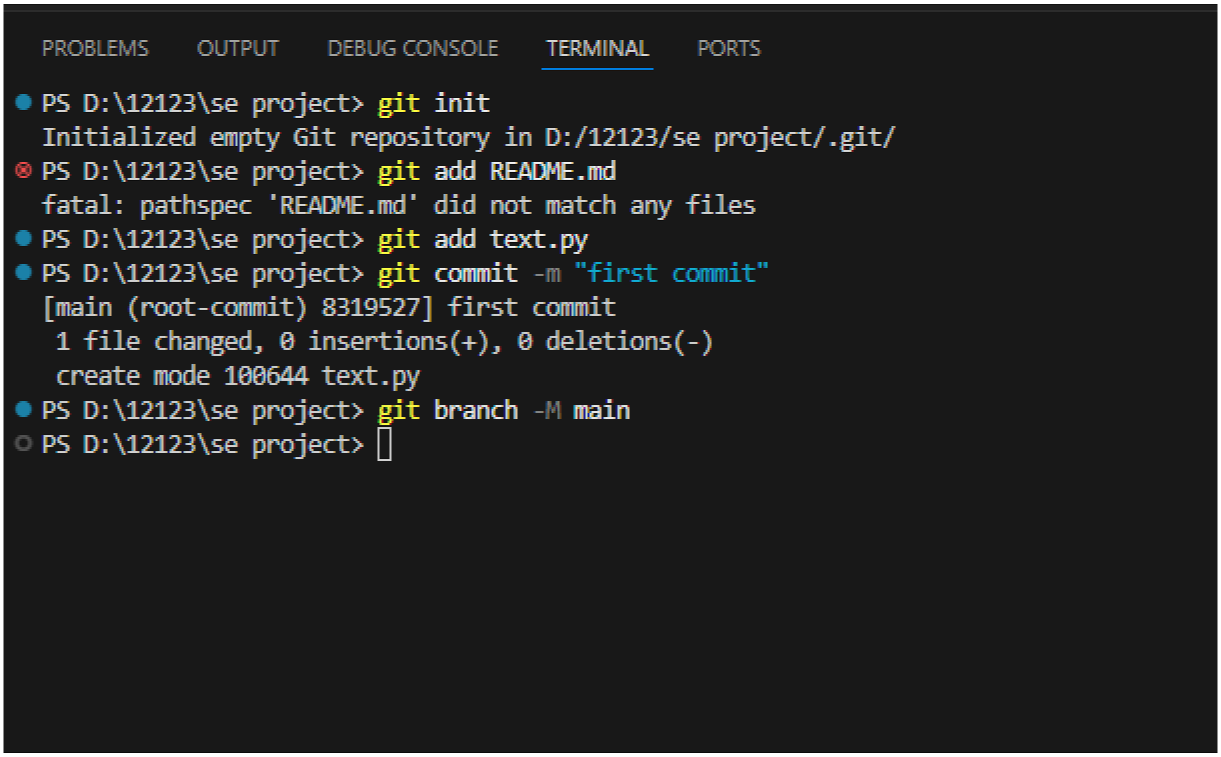
```
● PS D:\12123\se project> git init
  Initialized empty Git repository in D:/12123/se project/.git/
⊗ PS D:\12123\se project> git add README.md
  fatal: pathspec 'README.md' did not match any files
● PS D:\12123\se project> git add text.py
  PS D:\12123\se project> 
```



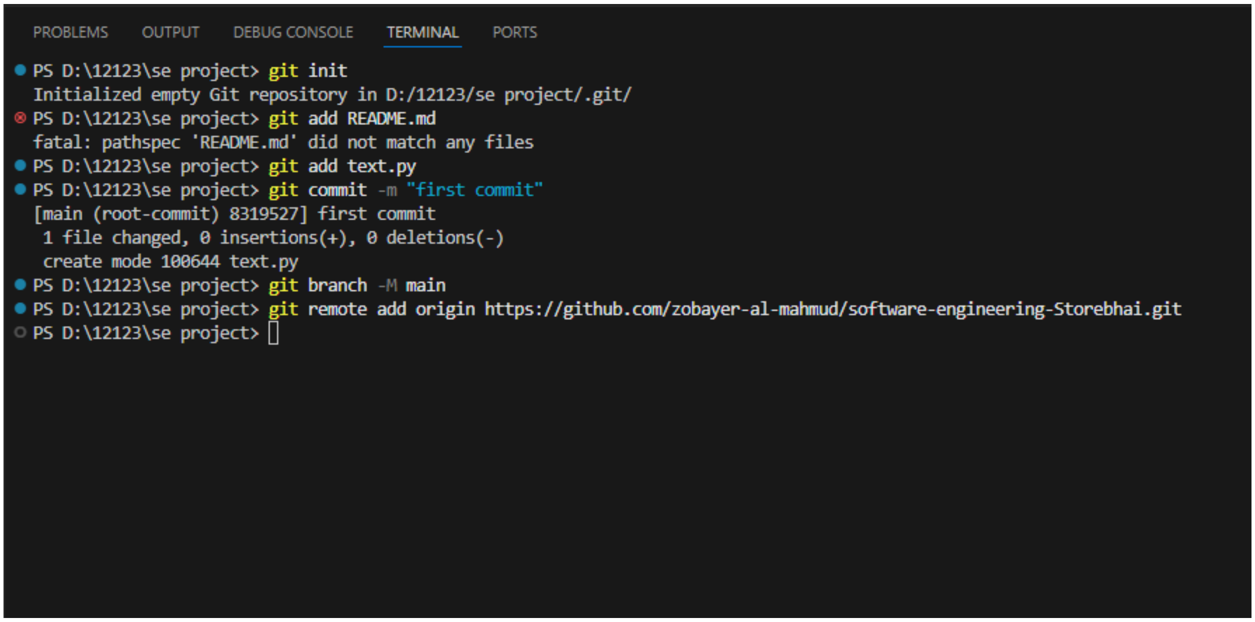
```
● PS D:\12123\se project> git init
  Initialized empty Git repository in D:/12123/se project/.git/
○ PS D:\12123\se project> 
```



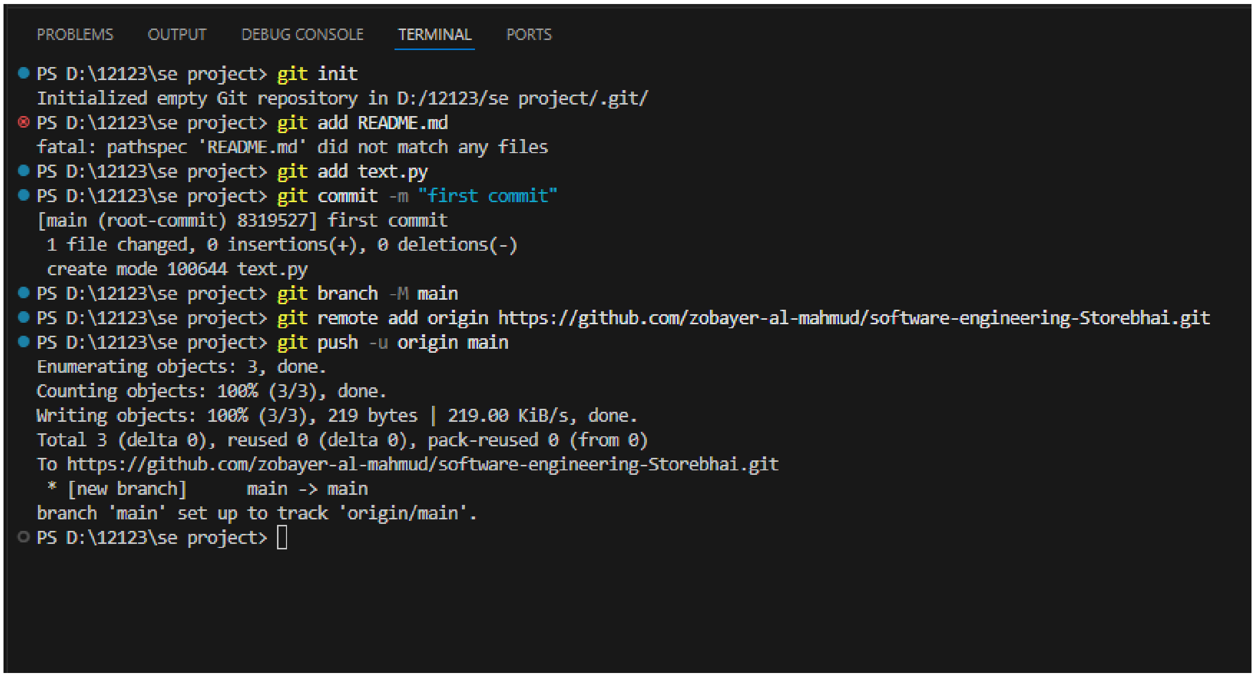
```
PS D:\12123\se project> git init
Initialized empty Git repository in D:/12123/se project/.git/
PS D:\12123\se project> git add README.md
fatal: pathspec 'README.md' did not match any files
PS D:\12123\se project> git add text.py
PS D:\12123\se project> git commit -m "first commit"
[main (root-commit) 8319527] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.py
PS D:\12123\se project> 
```



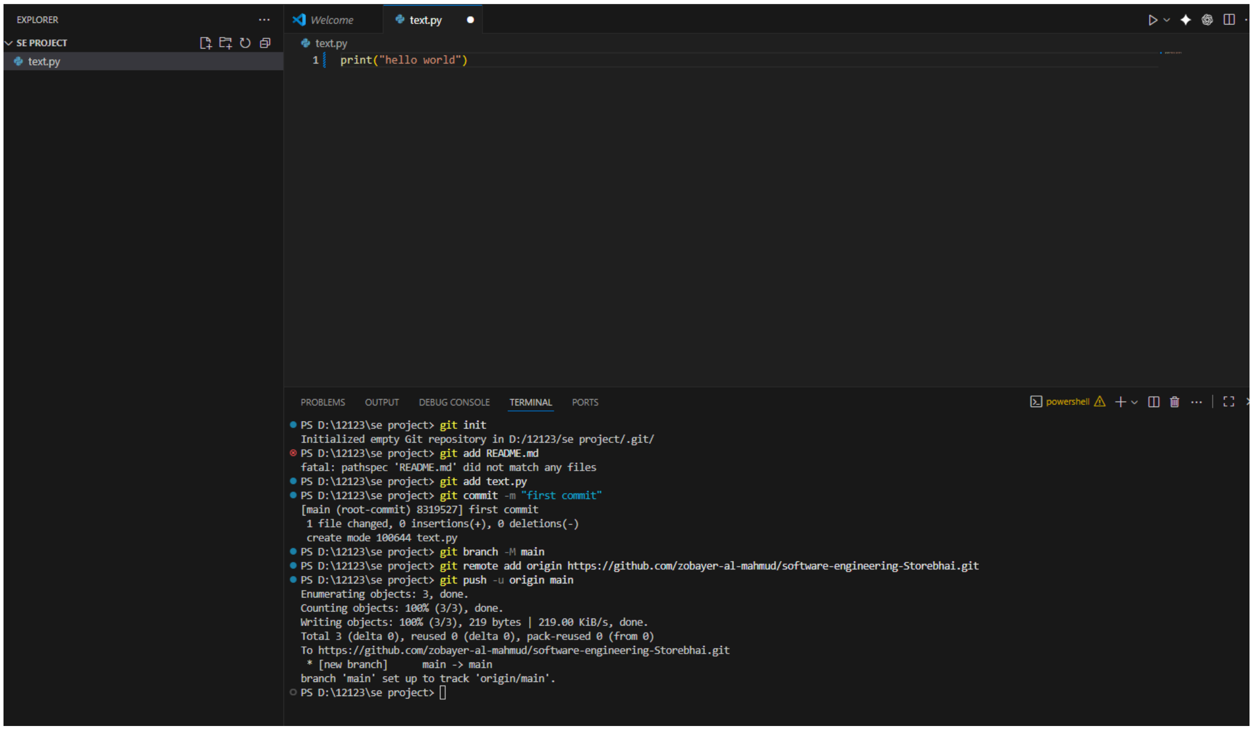
```
PS D:\12123\se project> git init
Initialized empty Git repository in D:/12123/se project/.git/
PS D:\12123\se project> git add README.md
fatal: pathspec 'README.md' did not match any files
PS D:\12123\se project> git add text.py
PS D:\12123\se project> git commit -m "first commit"
[main (root-commit) 8319527] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.py
PS D:\12123\se project> git branch -M main
PS D:\12123\se project> 
```



```
● PS D:\12123\se project> git init
  Initialized empty Git repository in D:/12123/se project/.git/
⊗ PS D:\12123\se project> git add README.md
  fatal: pathspec 'README.md' did not match any files
● PS D:\12123\se project> git add text.py
● PS D:\12123\se project> git commit -m "first commit"
  [main (root-commit) 8319527] first commit
   1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 text.py
● PS D:\12123\se project> git branch -M main
● PS D:\12123\se project> git remote add origin https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
○ PS D:\12123\se project> 
```

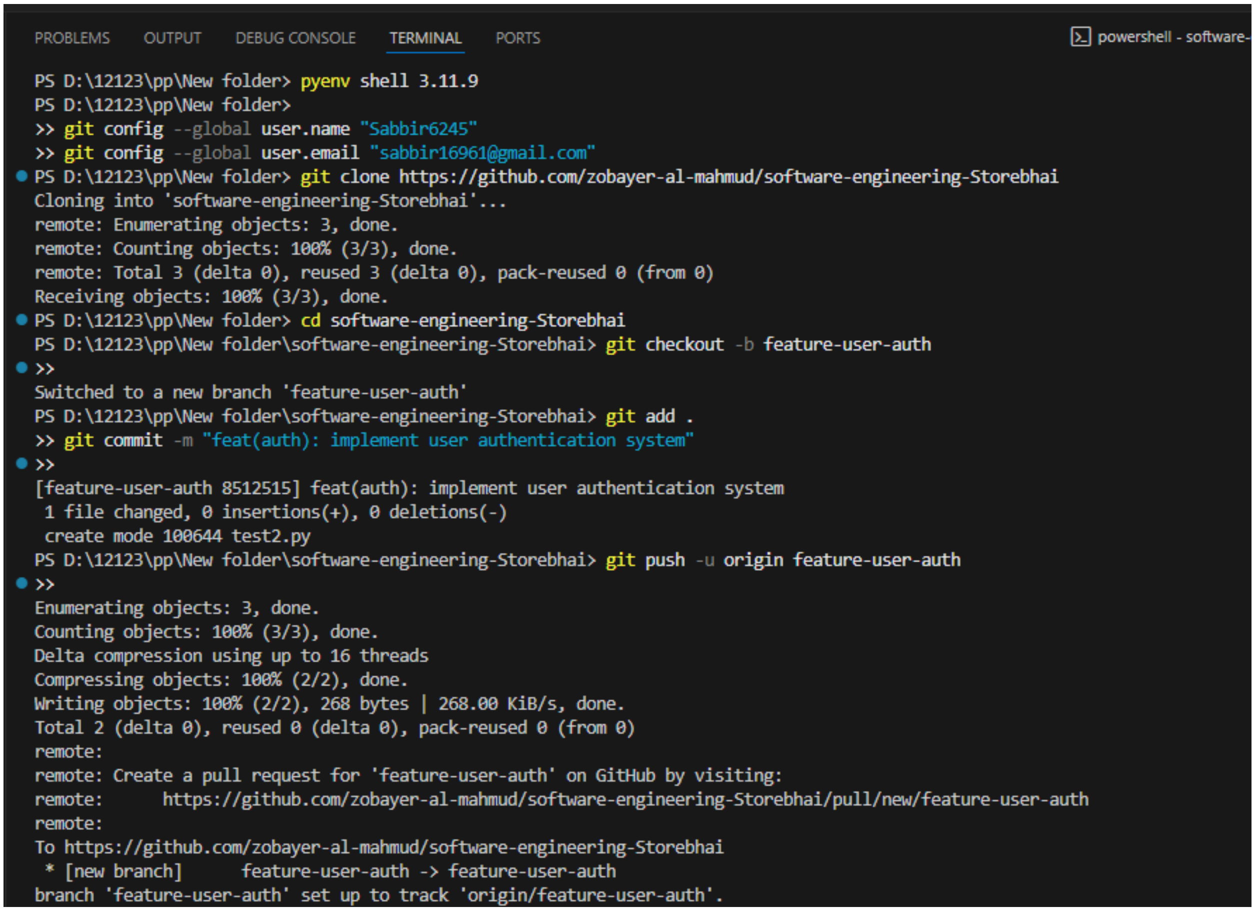


```
● PS D:\12123\se project> git init
  Initialized empty Git repository in D:/12123/se project/.git/
⊗ PS D:\12123\se project> git add README.md
  fatal: pathspec 'README.md' did not match any files
● PS D:\12123\se project> git add text.py
● PS D:\12123\se project> git commit -m "first commit"
  [main (root-commit) 8319527] first commit
   1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 text.py
● PS D:\12123\se project> git branch -M main
● PS D:\12123\se project> git remote add origin https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
● PS D:\12123\se project> git push -u origin main
  Enumerating objects: 3, done.
  Counting objects: 100% (3/3), done.
  Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
  Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
  To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
   * [new branch]      main -> main
  branch 'main' set up to track 'origin/main'.
○ PS D:\12123\se project> 
```

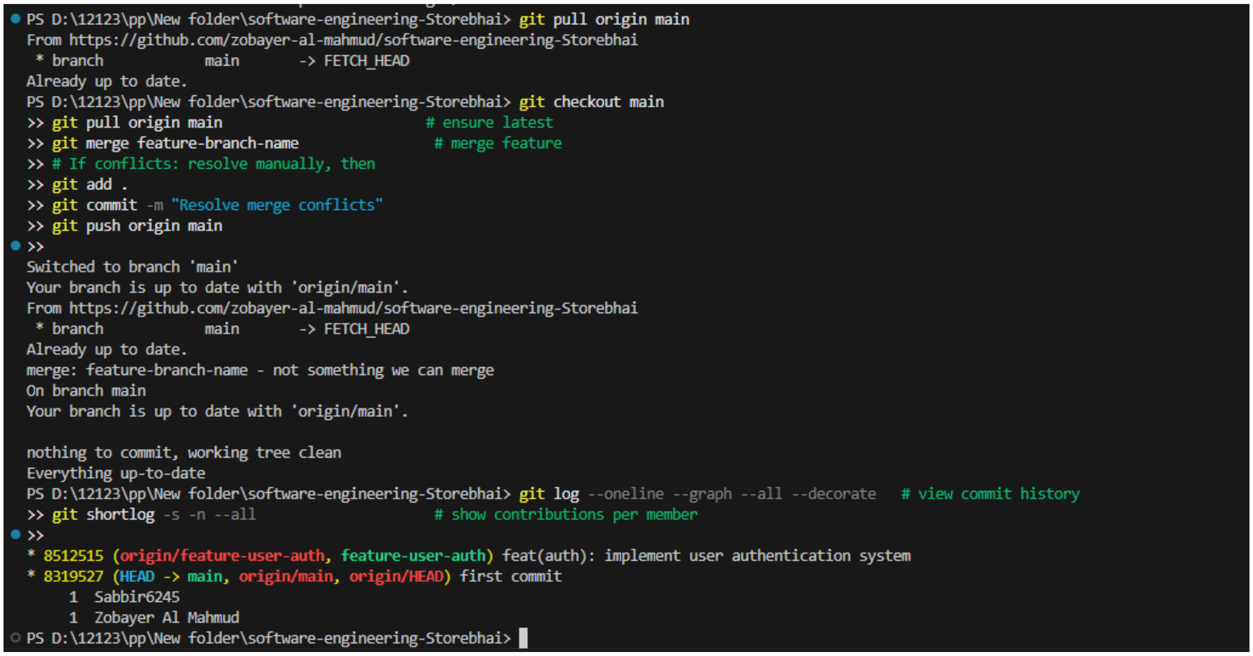


```python
print("hello world")
```

```
PS D:\12123\se project> git init
Initialized empty Git repository in D:/12123/se project/.git/
PS D:\12123\se project> git add README.md
fatal: pathspec 'README.md' did not match any files
PS D:\12123\se project> git add text.py
PS D:\12123\se project> git commit -m "first commit"
[main (root-commit) 8319527] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.py
PS D:\12123\se project> git branch -M main
PS D:\12123\se project> git remote add origin https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
PS D:\12123\se project> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS D:\12123\se project> 
```

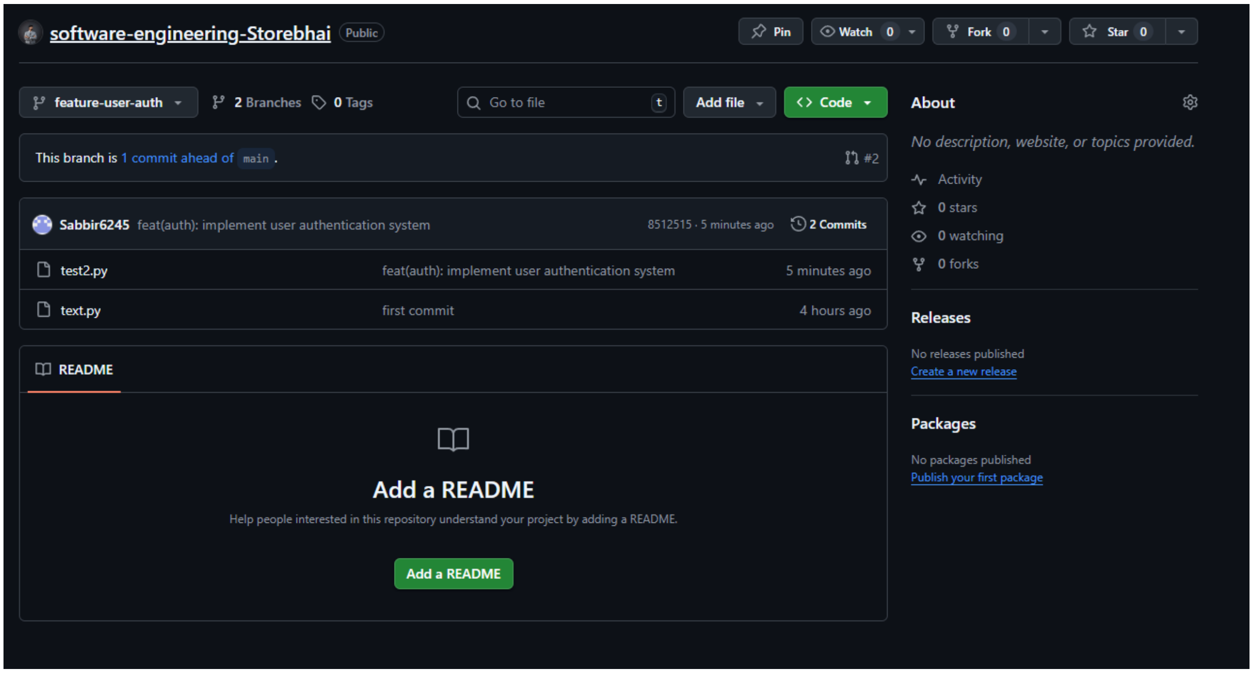


```
PS D:\12123\pp\New folder> pyenv shell 3.11.9
PS D:\12123\pp\New folder>
>> git config --global user.name "Sabbir6245"
>> git config --global user.email "sabbir16961@gmail.com"
PS D:\12123\pp\New folder> git clone https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
Cloning into 'software-engineering-Storebhai'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
PS D:\12123\pp\New folder> cd software-engineering-Storebhai
PS D:\12123\pp\New folder\software-engineering-Storebhai> git checkout -b feature-user-auth
>>
Switched to a new branch 'feature-user-auth'
PS D:\12123\pp\New folder\software-engineering-Storebhai> git add .
>> git commit -m "feat(auth): implement user authentication system"
>>
[feature-user-auth 8512515] feat(auth): implement user authentication system
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test2.py
PS D:\12123\pp\New folder\software-engineering-Storebhai> git push -u origin feature-user-auth
>>
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 268 bytes | 268.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-user-auth' on GitHub by visiting:
remote:      https://github.com/zobayer-al-mahmud/software-engineering-Storebhai/pull/new/feature-user-auth
remote:
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * [new branch]      feature-user-auth -> feature-user-auth
branch 'feature-user-auth' set up to track 'origin/feature-user-auth'.
```



```
PS D:\12123\pp\New folder\software-engineering-Storebhai> git pull origin main
From https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * branch            main       -> FETCH_HEAD
Already up to date.
PS D:\12123\pp\New folder\software-engineering-Storebhai> git checkout main
>> git pull origin main                              # ensure latest
>> git merge feature-branch-name                      # merge feature
>> # If conflicts: resolve manually, then
>> git add .
>> git commit -m "Resolve merge conflicts"
>> git push origin main
>>
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
From https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * branch            main       -> FETCH_HEAD
Already up to date.
merge: feature-branch-name - not something we can merge
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
PS D:\12123\pp\New folder\software-engineering-Storebhai> git log --oneline --graph --all --decorate   # view commit history
>> git shortlog -s -n --all                           # show contributions per member
>>
* 8512515 (origin/feature-user-auth, feature-user-auth) feat(auth): implement user authentication system
* 8319527 (HEAD -> main, origin/main, origin/HEAD) first commit
     1  Sabbir6245
     1  Zobayer Al Mahmud
PS D:\12123\pp\New folder\software-engineering-Storebhai>
```



software-engineering-Storebhai Public

This branch is 1 commit ahead of main . #2

Sabbir6245 feat(auth): implement user authentication system 8512515 · 5 minutes ago 2 Commits

test2.py	feat(auth): implement user authentication system	5 minutes ago
text.py	first commit	4 hours ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

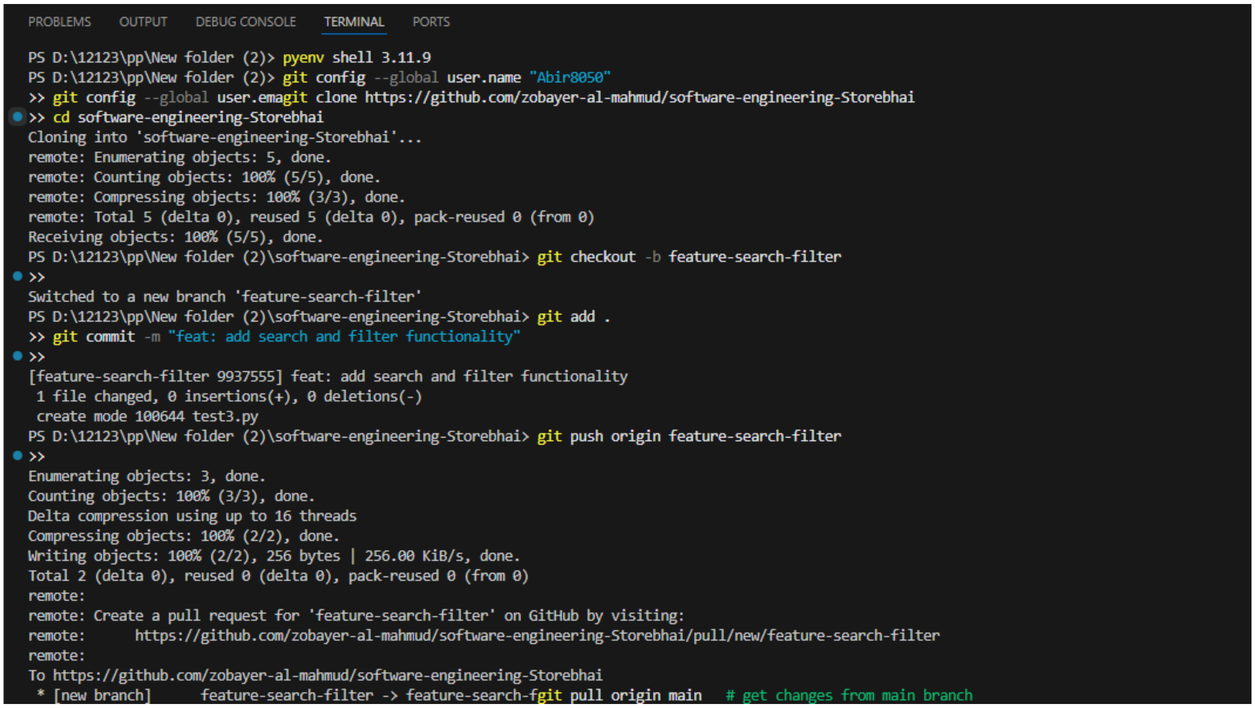
Activity
0 stars
0 watching
0 forks

Releases

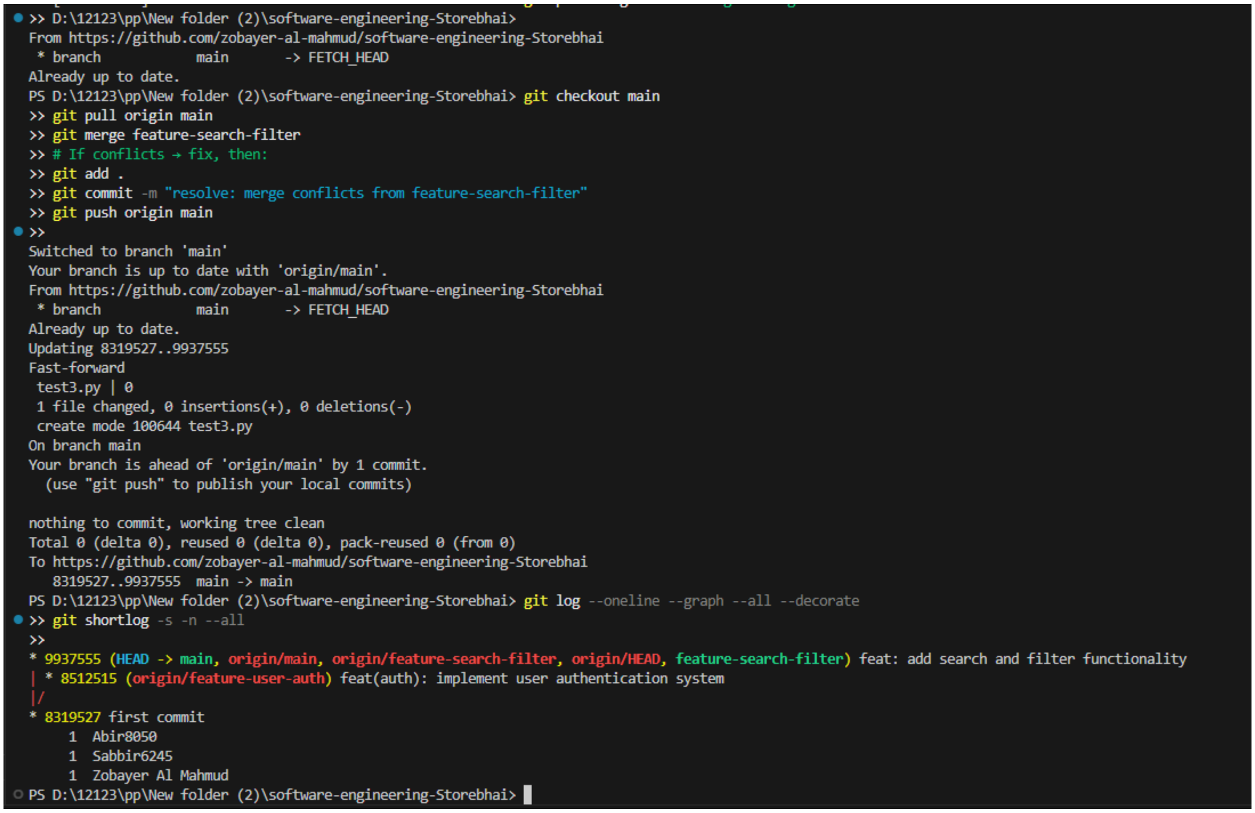
No releases published
Create a new release

Packages

No packages published
Publish your first package

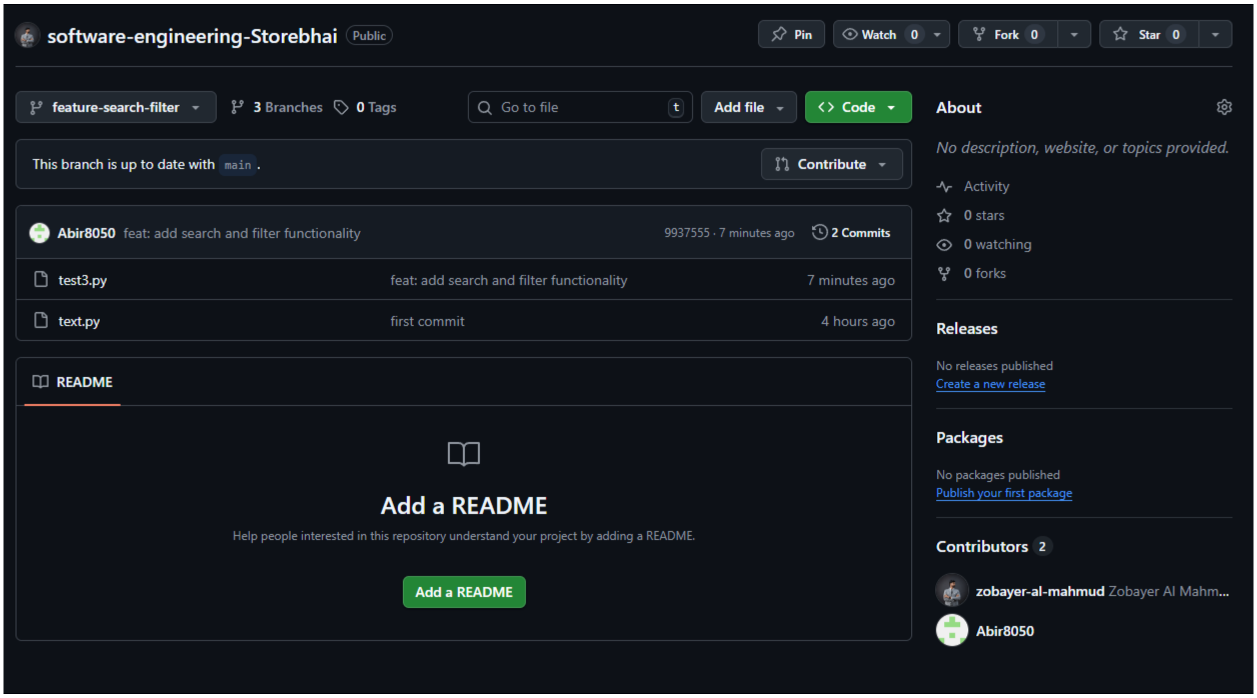


```
PS D:\12123\pp\New folder (2)> pyenv shell 3.11.9
PS D:\12123\pp\New folder (2)> git config --global user.name "Abir8050"
>> git config --global user.emagit clone https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
>> cd software-engineering-Storebhai
Cloning into 'software-engineering-Storebhai'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
PS D:\12123\pp\New folder (2)\software-engineering-Storebhai> git checkout -b feature-search-filter
>>
Switched to a new branch 'feature-search-filter'
PS D:\12123\pp\New folder (2)\software-engineering-Storebhai> git add .
>> git commit -m "feat: add search and filter functionality"
>>
[feature-search-filter 9937555] feat: add search and filter functionality
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test3.py
PS D:\12123\pp\New folder (2)\software-engineering-Storebhai> git push origin feature-search-filter
>>
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 256 bytes | 256.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-search-filter' on GitHub by visiting:
remote:      https://github.com/zobayer-al-mahmud/software-engineering-Storebhai/pull/new/feature-search-filter
remote:
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * [new branch]      feature-search-filter -> feature-search-fgit pull origin main   # get changes from main branch
```



```
>> D:\12123\pp\New folder (2)\software-engineering-Storebhai>
From https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * branch            main       -> FETCH_HEAD
Already up to date.
PS D:\12123\pp\New folder (2)\software-engineering-Storebhai> git checkout main
>> git pull origin main
>> git merge feature-search-filter
>> # If conflicts → fix, then:
>> git add .
>> git commit -m "resolve: merge conflicts from feature-search-filter"
>> git push origin main
>>
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
From https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * branch            main       -> FETCH_HEAD
Already up to date.
Updating 8319527..9937555
Fast-forward
 test3.py | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test3.py
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
   8319527..9937555  main -> main
PS D:\12123\pp\New folder (2)\software-engineering-Storebhai> git log --oneline --graph --all --decorate
>> git shortlog -s -n --all
>>
* 9937555 (HEAD -> main, origin/main, origin/feature-search-filter, origin/HEAD, feature-search-filter) feat: add search and filter functionality
| * 8512515 (origin/feature-user-auth) feat(auth): implement user authentication system
|/
* 8319527 first commit
     1  Abir8050
     1  Sabbir6245
     1  Zobayer Al Mahmud
PS D:\12123\pp\New folder (2)\software-engineering-Storebhai> 
```



This branch is up to date with `main`.

Abir8050 feat: add search and filter functionality 9937555 · 7 minutes ago 2 Commits

test3.py	feat: add search and filter functionality	7 minutes ago
text.py	first commit	4 hours ago

Help people interested in this repository understand your project by adding a README.

No description, website, or topics provided.

0 stars

0 watching

0 forks

No releases published

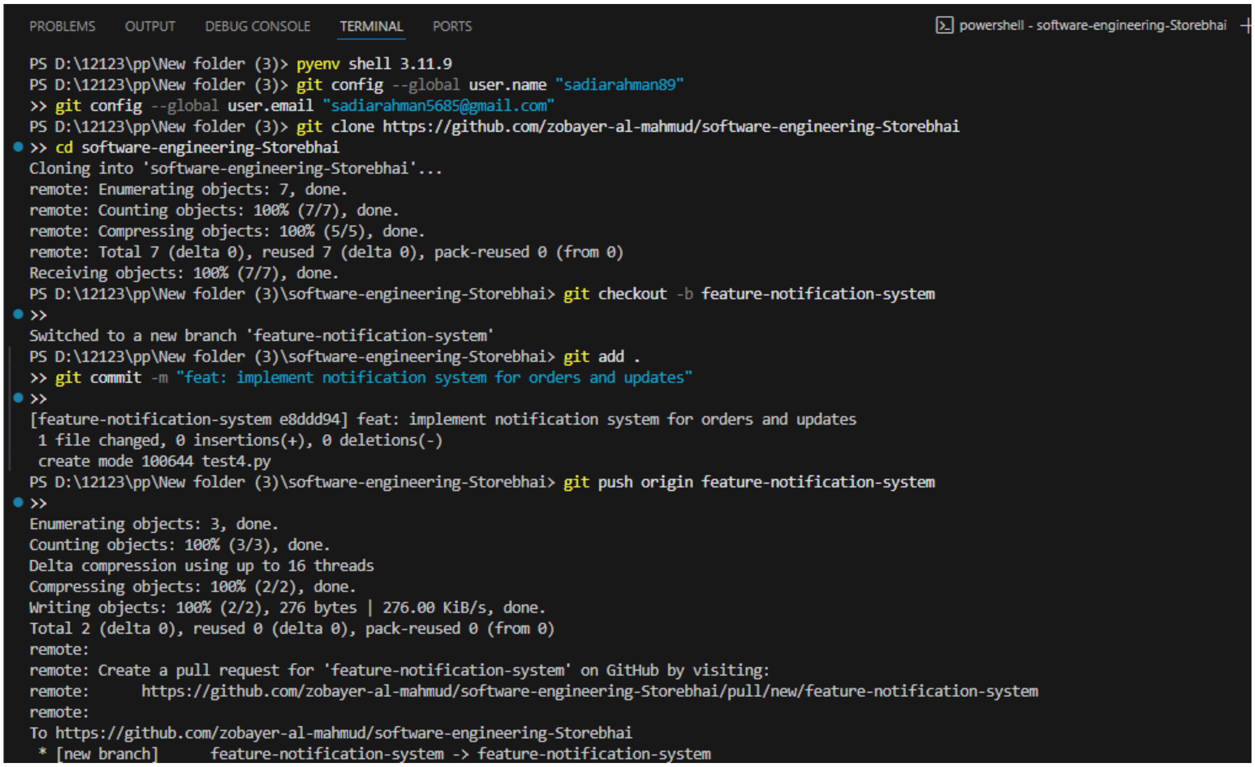
Create a new release

No packages published

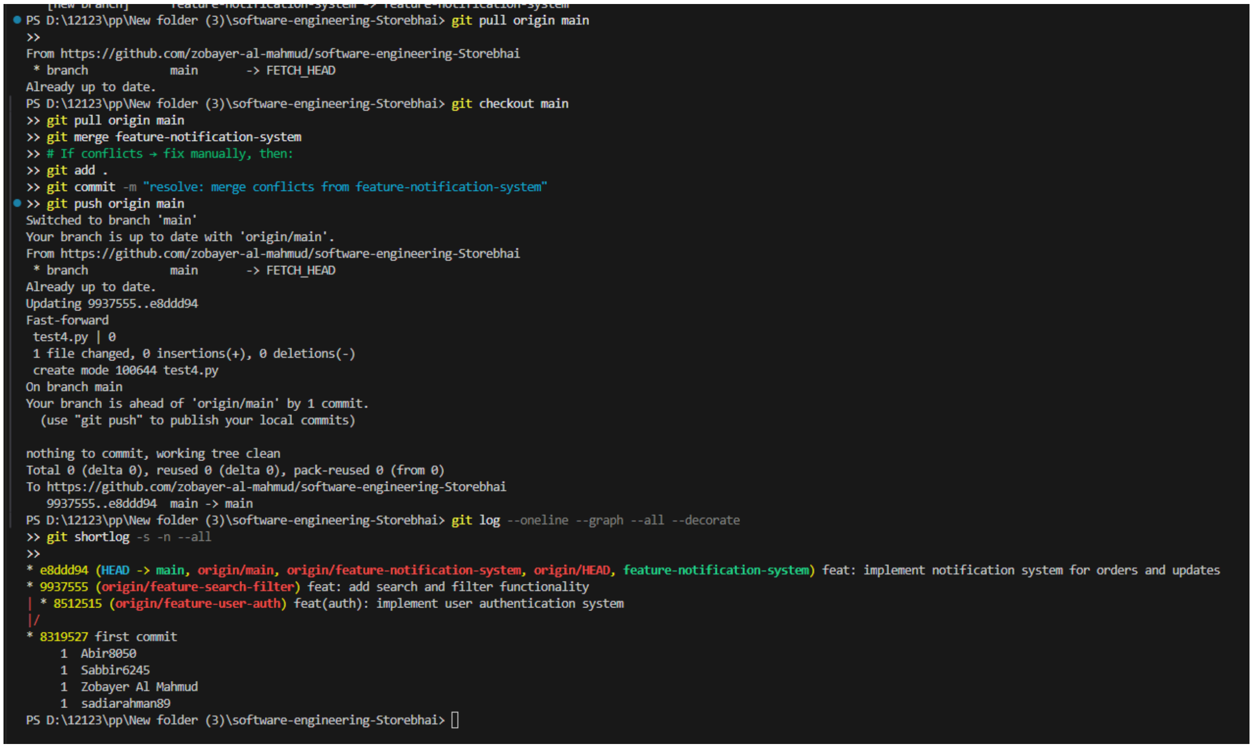
Publish your first package

zobayer-al-mahmud Zobayer Al Mahm...

Abir8050

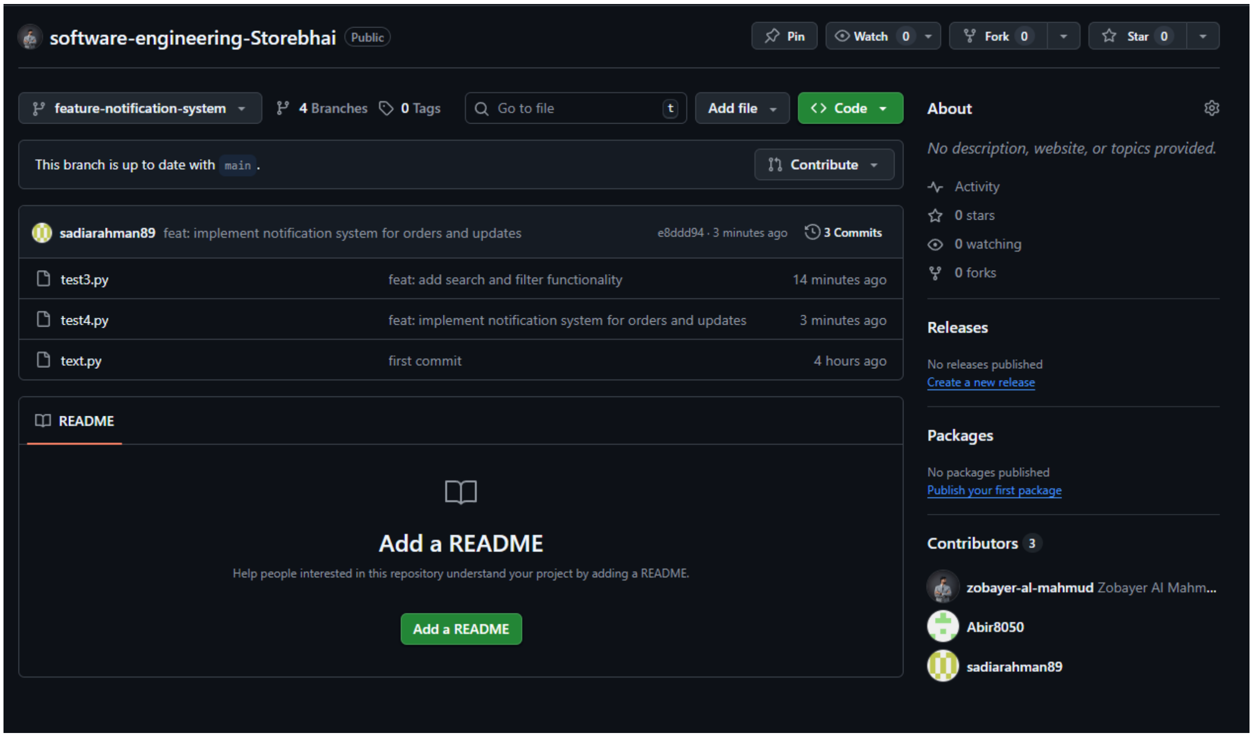


```
PS D:\12123\pp\New folder (3)> pyenv shell 3.11.9
PS D:\12123\pp\New folder (3)> git config --global user.name "sadiarahman89"
>> git config --global user.email "sadiarahman5685@gmail.com"
PS D:\12123\pp\New folder (3)> git clone https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
>> cd software-engineering-Storebhai
Cloning into 'software-engineering-Storebhai'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
PS D:\12123\pp\New folder (3)\software-engineering-Storebhai> git checkout -b feature-notification-system
>>
Switched to a new branch 'feature-notification-system'
PS D:\12123\pp\New folder (3)\software-engineering-Storebhai> git add .
>> git commit -m "feat: implement notification system for orders and updates"
>>
[feature-notification-system e8ddd94] feat: implement notification system for orders and updates
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test4.py
PS D:\12123\pp\New folder (3)\software-engineering-Storebhai> git push origin feature-notification-system
>>
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 276 bytes | 276.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-notification-system' on GitHub by visiting:
remote:      https://github.com/zobayer-al-mahmud/software-engineering-Storebhai/pull/new/feature-notification-system
remote:
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * [new branch]      feature-notification-system -> feature-notification-system
```



```
   [new branch]      feature-notification-system -> feature-notification-system
PS D:\12123\pp\New folder (3)\software-engineering-Storebhai> git pull origin main
>>
From https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * branch            main       -> FETCH_HEAD
Already up to date.
PS D:\12123\pp\New folder (3)\software-engineering-Storebhai> git checkout main
>> git pull origin main
>> git merge feature-notification-system
>> # If conflicts → fix manually, then:
>> git add .
>> git commit -m "resolve: merge conflicts from feature-notification-system"
>> git push origin main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
From https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * branch            main       -> FETCH_HEAD
Already up to date.
Updating 9937555..e8ddd94
Fast-forward
 test4.py | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test4.py
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
   9937555..e8ddd94  main -> main
PS D:\12123\pp\New folder (3)\software-engineering-Storebhai> git log --oneline --graph --all --decorate
>> git shortlog -s -n --all
>>
* e8ddd94 (HEAD -> main, origin/main, origin/feature-notification-system, origin/HEAD, feature-notification-system) feat: implement notification system for orders and updates
* 9937555 (origin/feature-search-filter) feat: add search and filter functionality
| * 8512515 (origin/feature-user-auth) feat(auth): implement user authentication system
|/
* 8319527 first commit
     1  Abir8050
     1  Sabbir6245
     1  Zobayer Al Mahmud
     1  sadiarahman89
PS D:\12123\pp\New folder (3)\software-engineering-Storebhai> 
```



This branch is up to date with main .

sadiarahman89 feat: implement notification system for orders and updates e8ddd94 · 3 minutes ago 3 Commits

test3.py	feat: add search and filter functionality	14 minutes ago
test4.py	feat: implement notification system for orders and updates	3 minutes ago
text.py	first commit	4 hours ago

Add a README

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

Releases

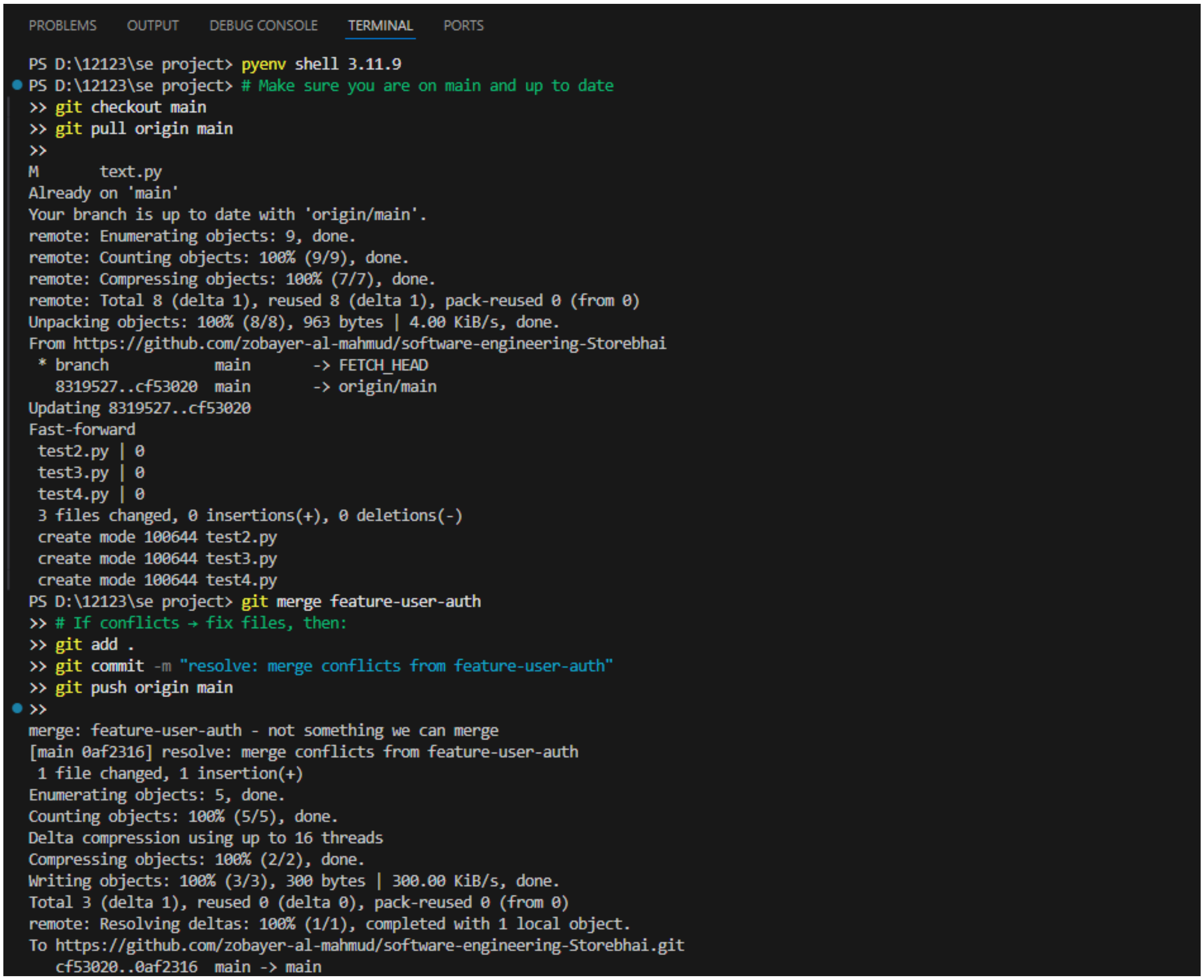
No releases published
Create a new release

Packages

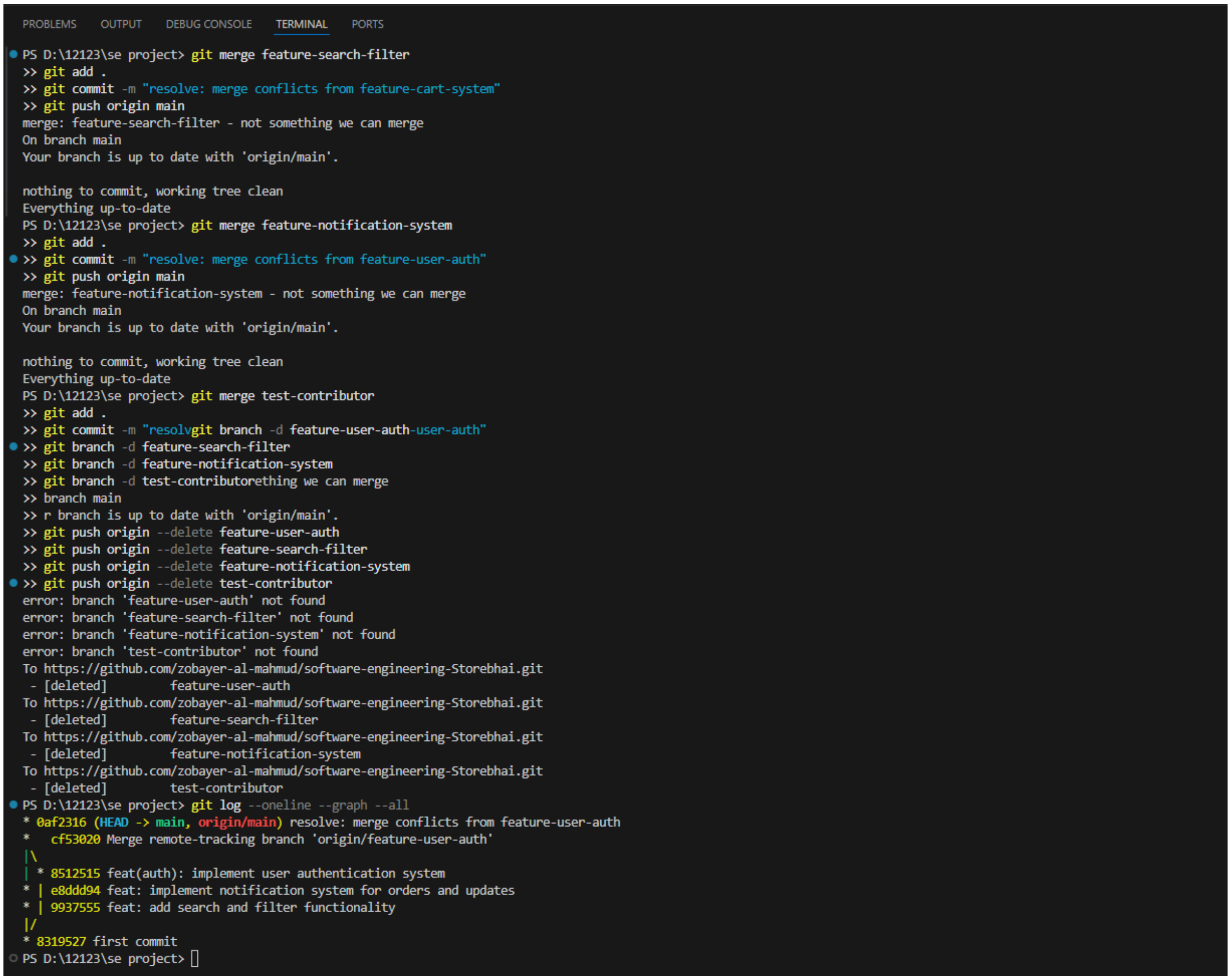
No packages published
Publish your first package

Contributors 3

zobayer-al-mahmud Zobayer Al Mahm...
Abir8050
sadiarahman89



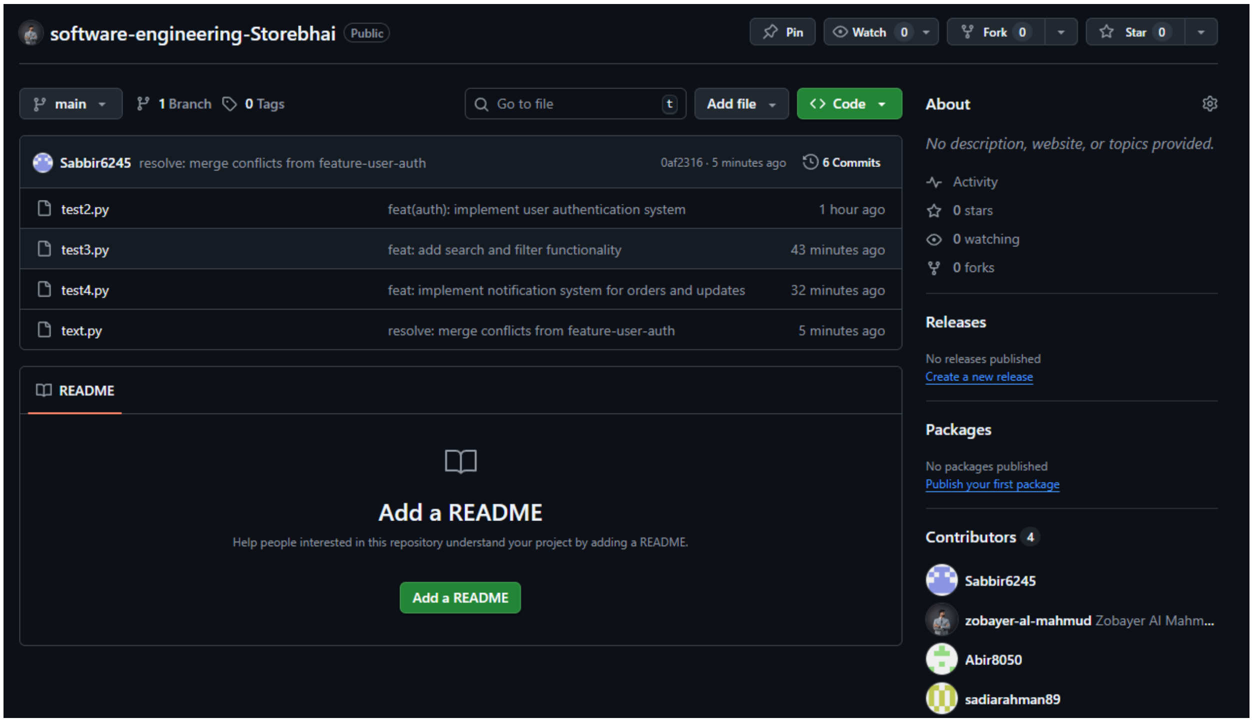
```
PS D:\12123\se project> pyenv shell 3.11.9
PS D:\12123\se project> # Make sure you are on main and up to date
>> git checkout main
>> git pull origin main
>>
M       text.py
Already on 'main'
Your branch is up to date with 'origin/main'.
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 8 (delta 1), pack-reused 0 (from 0)
Unpacking objects: 100% (8/8), 963 bytes | 4.00 KiB/s, done.
From https://github.com/zobayer-al-mahmud/software-engineering-Storebhai
 * branch            main       -> FETCH_HEAD
   8319527..cf53020  main       -> origin/main
Updating 8319527..cf53020
Fast-forward
 test2.py | 0
 test3.py | 0
 test4.py | 0
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test2.py
 create mode 100644 test3.py
 create mode 100644 test4.py
PS D:\12123\se project> git merge feature-user-auth
>> # If conflicts → fix files, then:
>> git add .
>> git commit -m "resolve: merge conflicts from feature-user-auth"
>> git push origin main
>>
merge: feature-user-auth - not something we can merge
[main 0af2316] resolve: merge conflicts from feature-user-auth
 1 file changed, 1 insertion(+)
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 300.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
   cf53020..0af2316  main -> main
```



```
PS D:\12123\se project> git merge feature-search-filter
>> git add .
>> git commit -m "resolve: merge conflicts from feature-cart-system"
>> git push origin main
merge: feature-search-filter - not something we can merge
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
PS D:\12123\se project> git merge feature-notification-system
>> git add .
>> git commit -m "resolve: merge conflicts from feature-user-auth"
>> git push origin main
merge: feature-notification-system - not something we can merge
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
PS D:\12123\se project> git merge test-contributor
>> git add .
>> git commit -m "resolvgit branch -d feature-user-auth-user-auth"
>> git branch -d feature-search-filter
>> git branch -d feature-notification-system
>> git branch -d test-contributorething we can merge
>> branch main
>> r branch is up to date with 'origin/main'.
>> git push origin --delete feature-user-auth
>> git push origin --delete feature-search-filter
>> git push origin --delete feature-notification-system
>> git push origin --delete test-contributor
error: branch 'feature-user-auth' not found
error: branch 'feature-search-filter' not found
error: branch 'feature-notification-system' not found
error: branch 'test-contributor' not found
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
 - [deleted]         feature-user-auth
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
 - [deleted]         feature-search-filter
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
 - [deleted]         feature-notification-system
To https://github.com/zobayer-al-mahmud/software-engineering-Storebhai.git
 - [deleted]         test-contributor
PS D:\12123\se project> git log --oneline --graph --all
* 0af2316 (HEAD -> main, origin/main) resolve: merge conflicts from feature-user-auth
*   cf53020 Merge remote-tracking branch 'origin/feature-user-auth'
|\
| * 8512515 feat(auth): implement user authentication system
* | e8ddd94 feat: implement notification system for orders and updates
* | 9937555 feat: add search and filter functionality
|/
* 8319527 first commit
PS D:\12123\se project> 
```



Sabbir6245 resolve: merge conflicts from feature-user-auth 0af2316 · 5 minutes ago 6 Commits

test2.py	feat(auth): implement user authentication system	1 hour ago
test3.py	feat: add search and filter functionality	43 minutes ago
test4.py	feat: implement notification system for orders and updates	32 minutes ago
text.py	resolve: merge conflicts from feature-user-auth	5 minutes ago

Add a README

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

Activity
0 stars
0 watching
0 forks

Releases

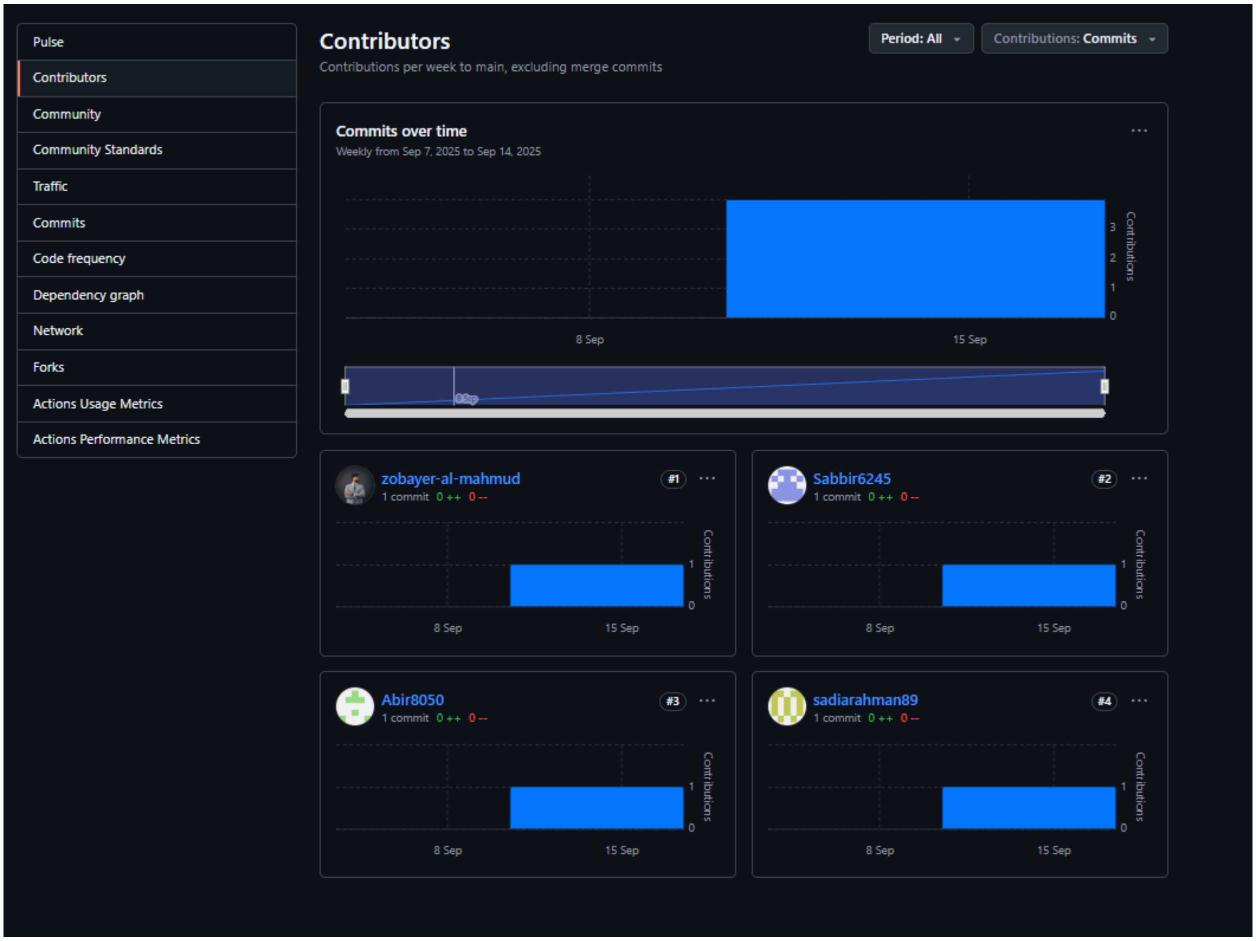
No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 4

Sabbir6245
zobayer-al-mahmud Zobayer Al Mahm...
Abir8050
sadiarahman89



Contributors

Contributions per week to main, excluding merge commits

Period: All

Contributions: Commits

Commits over time

Weekly from Sep 7, 2025 to Sep 14, 2025

zobayer-al-mahmud #1

1 commit 0 ++ 0 --

Sabbir6245 #2

1 commit 0 ++ 0 --

Abir8050 #3

1 commit 0 ++ 0 --

sadiarahman89 #4

1 commit 0 ++ 0 --

TESTING

Objective of Testing

The goal of testing is to ensure that Store Bhai meets all specified requirements and delivers a high-quality, reliable, and user-friendly platform. Testing will validate that all functionalities work as expected, user inputs are correctly processed, and the system handles errors gracefully. By conducting systematic testing, the team aims to minimize defects, ensure data integrity, and confirm that the platform conforms to business and technical requirements.

Testing Methods

For Store Bhai, the following testing methods will be used:

1. **Unit Testing**: Each module or component, such as the login system, booking system, or payment integration, will be tested independently to ensure correct functionality.
2. **System Testing:** The complete system will be tested to confirm that it meets all functional and non-functional requirements**.**
3. **User Acceptance Testing (UAT**): End-users, such as shop owners and renters, will test the system to ensure it is intuitive, meets real-world needs, and is ready for deployment.
4. **Regression Testing:** After updates or bug fixes, tests will be repeated to ensure that existing functionality remains unaffected.
5. **Security Testing:** Tests will check that user authentication, data privacy, and payment security meet required standards.

Sabbir's Test Cases (TC_1 – TC_4)

TC_1

Project Name: StoreBhai Test Designed by: Sabbir

Test Case ID: TC_1 Test Designed date: 25/08/2025

Test Priority: High Test Executed by: Sabbir

Module Name: Authentication → Signup Test Execution date: 28/08/2025

Test Title: Signup (Host) with valid details

Description: Positive path for Host registration with unique phone/email and valid inputs.

Precondition: Phone and email not registered; system online.
Dependencies: OTP/email service available

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Open Signup page 2. Enter Name 3. Enter unique Phone/Email 4. Enter Password (≥8 chars) 5. Re-enter Password 6. Select User Type = Host 7. Click Signup	Name: Arif Hasan Phone: 01711000001 Email: arif.host@example.com Password: Abcd1234! Re-password: Abcd1234! User Type: Host	Account created successfully Confirmation SMS/email sent Redirect to Host Dashboard	As expected	Pass

---

TC_2
Project Name: StoreBhai Test Designed by: Sabbir
Test Case ID: TC_2 Test Designed date: 27/08/2025
Test Priority: High Test Executed by: Sabbir
Module Name: Authentication → Signup Test Execution date: 29/08/2025
Test Title: Duplicate signup attempt
Description: Prevent duplicate signup when phone/email already exists.
Precondition: Account already registered with given phone/email.
Dependencies: Database seeded with existing user.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Signup 2. Enter existing phone/email 3. Fill other valid details 4. Click Signup	Phone: 01711000002 Email: existing@example.com Password: Abcd1234!	Error: “Account already exists.” No account created	As expected	Pass

TC_3
Project Name: StoreBhai Test Designed by: Sabbir
Test Case ID: TC_3 Test Designed date: 30/08/2025
Test Priority: Medium Test Executed by: Sabbir
Module Name: Authentication → Signup Test Execution date: 01/09/2025
Test Title: Password too short
Description: Password must be at least 8 characters.
Precondition: Unique phone/email.
Dependencies: None

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Signup 2. Enter password < 8 chars	Password: Abc123	Error: “Password too short.” Signup blocked	As expected	Pass

Test Steps	Test Data	Expected Results	Actual Results	Status
3. Click Signup				

TC_4
Project Name: StoreBhai Test Designed by: Sabbir
Test Case ID: TC_4 Test Designed date: 02/09/2025
Test Priority: Medium Test Executed by: Sabbir
Module Name: Authentication → Signup Test Execution date: 03/09/2025
Test Title: Password mismatch validation
Description: Re-entered password must match original.
Precondition: Unique phone/email.
Dependencies: None

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Signup 2. Enter valid details 3. Enter different Password and Re-password 4. Click Signup	Password: Abcd1234! Re-password: Abcd1235!	Error: “Passwords do not match.” Signup blocked	As expected	Pass

Zobayer’s Test Cases (TC_5 – TC_8)
TC_5
Project Name: StoreBhai Test Designed by: Zobayer
Test Case ID: TC_5 Test Designed date: 03/09/2025
Test Priority: High Test Executed by: Zobayer
Module Name: Authentication → Signin Test Execution date: 04/09/2025
Test Title: Valid login
Description: Successful login with correct details.
Precondition: Account exists and is active.
Dependencies: Session service active.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Signin 2. Enter valid credentials 3. Click Signin	Phone: 01711000003 Password: Abcd1234!	Redirect to Dashboard Session created	As expected	Pass

TC_6
Project Name: StoreBhai Test Designed by: Zobayer
Test Case ID: TC_6 Test Designed date: 04/09/2025
Test Priority: High Test Executed by: Zobayer
Module Name: Authentication → Signin Test Execution date: 05/09/2025
Test Title: Wrong password
Description: Show error message on wrong password.
Precondition: Valid account exists.
Dependencies: None.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Signin 2. Enter wrong password 3. Click Signin	Phone: 01711000004 Password: WrongPass	Error: "Incorrect password." No session created	As expected	Pass

TC_7
Project Name: StoreBhai Test Designed by: Zobayer
Test Case ID: TC_7 Test Designed date: 05/09/2025
Test Priority: Medium Test Executed by: Zobayer
Module Name: Authentication → Signin Test Execution date: 06/09/2025
Test Title: Suspended account login
Description: Suspended users cannot log in.
Precondition: Account status = Suspended.
Dependencies: Admin tool to suspend account.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Signin 2. Enter credentials of suspended user 3. Click Signin	Phone: 01711000005 Password: Abcd1234!	Error: "Account inactive. Contact support."	As expected	Pass

---

TC_8
Project Name: StoreBhai Test Designed by: Zobayer
Test Case ID: TC_8 Test Designed date: 06/09/2025
Test Priority: High Test Executed by: Zobayer
Module Name: Forgot Password Test Execution date: 07/09/2025
Test Title: Valid OTP reset
Description: Password reset works with valid OTP within 10 minutes.
Precondition: Account exists; OTP service working.
Dependencies: OTP gateway.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Forgot Password 2. Enter email/phone 3. Enter valid OTP 4. Enter new password 5. Submit	Email: user@example.com OTP: 654321 New Password: NewPass123!	Password updated Redirect to Signin	As expected	Pass

Abir's Test Cases (TC_9 – TC_12)
TC_9
Project Name: StoreBhai Test Designed by: Abir
Test Case ID: TC_9 Test Designed date: 07/09/2025
Test Priority: Medium Test Executed by: Abir
Module Name: Forgot Password Test Execution date: 08/09/2025
Test Title: Expired OTP
Description: Expired or wrong OTP is rejected.

Precondition: Account exists.
Dependencies: OTP expiry rule enforced (10 min).

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Forgot Password 2. Enter expired OTP 3. Submit reset	OTP: 111111 (expired)	Error: “Invalid code.” Password not changed	As expected	Pass

TC_10
Project Name: StoreBhai Test Designed by: Abir
Test Case ID: TC_10 Test Designed date: 08/09/2025
Test Priority: Medium Test Executed by: Abir
Module Name: Dashboard Menu (Renter) Test Execution date: 09/09/2025
Test Title: Renter menu navigation
Description: Renter menu items route correctly.
Precondition: Logged in as Renter.
Dependencies: Routing enabled.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Login as Renter 2. Verify menu items visible 3. Click each item 4. Click Logout	Menu: Dashboard, Search Space, My Bookings, Payments, Messages, Profile, Help & Support, Logout	Each link opens correct page Logout ends session	As expected	Pass

TC_11
Project Name: StoreBhai Test Designed by: Abir
Test Case ID: TC_11 Test Designed date: 09/09/2025
Test Priority: High Test Executed by: Abir
Module Name: Admin Panel Test Execution date: 10/09/2025
Test Title: Admin menu authorization
Description: Admin-only items hidden from non-admins; links open correct sections.
Precondition: Admin and non-admin accounts exist.
Dependencies: Role-based access control.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Login as Admin and verify items 2. Login as Renter/Host and verify Admin items hidden	Admin Menu: Dashboard, Manage Users, Manage Listings, Manage Bookings, Payments & Payouts, Disputes, Reports, Settings, Profile, Logout	Admins see all items with correct routing Non-admins cannot see Admin items	As expected	Pass

TC_12
Project Name: StoreBhai Test Designed by: Abir
Test Case ID: TC_12 Test Designed date: 10/09/2025
Test Priority: Medium Test Executed by: Abir

Module Name: Top Bar Test Execution date: 11/09/2025
Test Title: Top bar functions
Description: Logo, Search, Notifications, Profile dropdown work consistently.
Precondition: Logged in; at least one notification exists.
Dependencies: Search and notification services.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Click Logo 2. Use Search 3. Open Notifications 4. Open Profile dropdown	Search example: "space in Dhaka"	Logo → Dashboard Search works Notifications show unread Profile dropdown opens	As expected	Pass

2. Sadia's Test Cases (TC_13 – TC_16)
3. TC_13

Project Name: StoreBhai Test Designed by: Sadia
Test Case ID: TC_13 Test Designed date: 11/09/2025
Test Priority: Low Test Executed by: Sadia
Module Name: Bottom Bar Test Execution date: 12/09/2025
Test Title: Bottom bar info and links
Description: Copyright, legal links, and version visible.
Precondition: Any logged-in user.
Dependencies: Terms, Privacy, Help pages exist.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Check copyright year 2. Click Terms, Privacy, Help 3. Verify version label	—	Year matches server current year Links open correct pages Version label present	As expected	Pass

TC_14
Project Name: StoreBhai Test Designed by: Sadia
Test Case ID: TC_14 Test Designed date: 12/09/2025
Test Priority: Medium Test Executed by: Sadia
Module Name: Profile → View Profile Test Execution date: 13/09/2025
Test Title: Profile view fields
Description: Profile shows saved data correctly.
Precondition: User has Name, Phone, Email, DOB, Gender, User Type saved.
Dependencies: DB read API.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open View Profile	—	All fields displayed correctly Edit button available	As expected	Pass

TC_15
Project Name: StoreBhai Test Designed by: Sadia
Test Case ID: TC_15 Test Designed date: 13/09/2025
Test Priority: High Test Executed by: Sadia

Module Name: Profile → Edit Profile   Test Execution date: 14/09/2025
Test Title: Edit Profile with valid data
Description: Update profile details successfully.
Precondition: Logged in; existing profile data.
Dependencies: Validation rules, DB update API.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Edit Profile 2. Change phone, email, DOB, gender 3. Click Confirm 4. Refresh and re-open	Phone: 01711000009 Email: new.mail@example.com DOB: 1997-05-10 Gender: Male	Success message “Profile updated” Values persist after refresh	As expected	Pass

TC_16
Project Name: StoreBhai   Test Designed by: Sadia
Test Case ID: TC_16   Test Designed date: 14/09/2025
Test Priority: Medium   Test Executed by: Sadia
Module Name: Profile → Change Photo   Test Execution date: 15/09/2025
Test Title: Upload new photo
Description: Only image files allowed; profile shows new photo after upload.
Precondition: Logged in.
Dependencies: File upload service; image validation.

Test Steps	Test Data	Expected Results	Actual Results	Status
1. Open Change Photo 2. Upload invalid file 3. Upload valid .jpg/.png 4. Click Upload	A) doc.pdf (invalid) B) avatar.jpg (valid)	A) Error about allowed formats B) Success; new photo visible on profile	As expected	Pass

SOFTWARE PRODUCT METRICS

The goal of applying software product metrics is to measure and assess the quality, maintainability, and complexity of the Store Bhai system. Using object-oriented and class metrics, we can evaluate the design’s structure and ensure it meets software engineering standards.

Object-Oriented and Class Metrics:

1. **Size Metrics**
    - Number of Classes (NOC): Store Bhai contains 7 primary classes, including User, Renter, Host, Admin, Listing, Booking, Payment, and Payout. A moderate number of classes ensures modularity and maintainability.
    - Lines of Code (LOC): The estimated size of the codebase is approximately 15,000 lines, split between frontend and backend. This metric helps predict effort, complexity, and testing requirements.
2. **Complexity Metrics**
    - Cyclomatic Complexity (CC): Each module, such as authentication, booking, and payment processing, is assessed for branching complexity. Complexity is kept low to medium to ensure easier testing and maintenance.
    - Weighted Methods per Class (WMC): The average WMC is calculated to estimate how complex each class is based on its number of methods. Classes like Booking and Payment have higher WMC due to multiple operations, while User-related classes remain simpler.
3. **Coupling Metrics**
    - Coupling Between Objects (CBO): Classes interact minimally, with Booking and Payment interacting with User and StorageSpace. Low coupling is maintained to reduce dependency and improve reusability.
    - Response for a Class (RFC): RFC measures the number of methods that can be invoked in response to a message. For example, the Renter class has methods such as SearchSpace(), BookSpace(), and MakePayment(), resulting in moderate RFC.
4. **Cohesion Metrics**
    - Lack of Cohesion of Methods (LCOM): Each class is designed with high cohesion. For example, the StorageSpace class handles only attributes and operations related to storage listing, ensuring responsibilities are focused and cohesive.
    - Tight cohesion ensures easier maintainability, understandability, and less chance of errors during modification.
5. **Inheritance Metrics**
    - Depth of Inheritance Tree (DIT): The User class serves as the base class for Renter, Host, and Admin, giving a DIT of 1 for derived classes. This shallow hierarchy reduces complexity while allowing specialization.
    - Number of Children (NOC): The User class has three child classes, which allows polymorphism without increasing complexity.
    - Coupling Between Classes (CBC): Inherited relationships are minimal, reducing dependency between parent and child classes.

Analysis and Benefits:

- By monitoring size, complexity, coupling, cohesion, and inheritance metrics, we can ensure that the Store Bhai system is maintainable, modular, and easy to extend.
- These metrics allow early detection of design weaknesses, such as excessive coupling or low cohesion, which could lead to increased maintenance costs or bugs.
- The system's moderate class size, high cohesion, and low coupling make it robust and scalable, supporting future enhancements such as loyalty programs, advanced reporting, or additional payment integrations.

CONCLUSION AND FUTURE WORK

Conclusion

The Store Bhai project successfully demonstrates the design and development of a peer-to-peer storage rental platform tailored for the needs of small and medium shop owners, students, drivers, and other users in Bangladesh. By leveraging a web-based system with secure user authentication, location-based search, booking, payment integration, and a robust admin panel, the platform addresses the problem of fragmented, manual storage management.

The application of the Scrum software development model ensured that the project was developed iteratively with continuous stakeholder feedback, resulting in a product that is both user-friendly and aligned with business objectives. The implementation of object-oriented design principles and class metrics, including size, complexity, coupling, cohesion, and inheritance measures, confirmed that the system is maintainable, modular, and scalable.

Overall, Store Bhai enhances operational efficiency, increases transparency, and provides a reliable, secure, and convenient solution for both hosts and renters. The project also contributes to scientific and professional development by documenting iterative improvements, testing cycles, and backlog management, serving as a reference for future peer-to-peer platform implementations.

Future Work

There are several opportunities to expand and improve Store Bhai in the future:

1. **Mobile Application Development**: Develop native Android and iOS apps to increase accessibility for users on the go.
2. **Advanced Search and Filtering**: Integrate features such as price ranges, storage type, security level, and distance optimization to enhance user experience.
3. **Loyalty Programs and Promotions**: Introduce reward points, discounts, or referral systems to increase user engagement and retention.
4. **Integration with Additional Payment Systems**: Expand beyond bKash and Nagad to include other local and international payment gateways.
5. **Automated Recommendation System**: Implement machine learning algorithms to suggest storage spaces based on user preferences and past bookings.
6. **Analytics and Reporting for Hosts**: Provide hosts with detailed insights on occupancy rates, revenue trends, and customer feedback to help them manage their spaces more effectively.
7. **Enhanced Security Features**: Introduce two-factor authentication, identity verification, and fraud detection systems to ensure maximum trust and safety.
8. **Regional Expansion**: Scale the platform to other cities in Bangladesh and eventually to other countries with similar storage needs.

By addressing these future enhancements, Store Bhai can evolve into a fully-featured, comprehensive storage management platform that meets the growing demands of users and continues to deliver real-world value.