

Smart Budget								
Team FTX								
7/23/23								
Scenario #	Test Scenario	Prerequisite	Test Description	Steps	Test Data	Expected Behavior	Actual Behavior	Result
1	Validation of ability for user to register an account	Server is running	Register with blank fields	1. Launch App 2. Click "Sign Up" 3. Enter fields 4. Click "Register"	All fields valid except Last Name:	Unsuccessful Registration	Unsuccessful Registration	Pass
2			Register with password shorter than 6		All fields valid except Password: 12312	Unsuccessful Registration	Unsuccessful Registration	Pass
3			Register with no matching confirm password		All fields valid except Password: 123456 Confirm Password: 123457	Unsuccessful Registration	Unsuccessful Registration	Pass
4			Register with no valid email		All fields valid except Email: poopybutt	Unsuccessful Registration	Unsuccessful Registration	Pass
5			Register with existing username		Other username: Bobby New username: Bobby	Unsuccessful Registration	Unsuccessful Registration	Pass
6			Register with all valid fields		All fields valid	Successful Registration	Successful Registration	Pass
7	Validation of ability for user to login to their account	1. Server is running 2. Pre-existing user account (Username: bob, Password: hill123)	Login with invalid username	1. Launch App 2. Enter username 3. Enter password 4. Click "Login"	Username: bobby Password: hill123	Unsuccessful Login	Unsuccessful Login	Pass
8			Login with invalid password		Username: bob Password: hill321	Unsuccessful Login	Unsuccessful Login	Pass
9			Login with valid username and password		Username: bob password: hill123	Successful Login	Successful Login	Pass
10	Validation of ability for user to view user's profile info	1. Server is running 2. Pre-existing user account	Login and check profile page	1. Launch App 2. Login 3. Click "Profile"	None	See profile info	See profile info	Pass
11	Validation of ability for user to logout from their account	1. Server is running 2. Pre-existing user account	Logout and check profile directory	1. Launch App 2. Login 3. Click "Profile" 4. Click "Logout" 5. Check directory	Check <a href="http://localhost:3000/profile">http://localhost:3000/profile</a>	No user info shown	No user info shown	Pass
12			Logout and check dashboard directory		Check <a href="http://localhost:3000/dashboard">http://localhost:3000/dashboard</a>	No user info shown	No user info shown	Pass
13			Logout and check savings directory		Check <a href="http://localhost:3000/savings">http://localhost:3000/savings</a>	No user info shown	No user info shown	Pass

14			Logout and check management directory		Check <a href="http://localhost:3000/add">http://localhost:3000/add</a>	No user info shown	No user info shown	Pass
15	Validation of ability for user to add saving goals	1. Server is running 2. Pre-existing user account	Add goal with missing field	1. Launch App 2. Click "Savings" 3. Click "Add Saving Goal" 4. Enter Goal Name 5. Enter Total 6. Enter Payment	Goal Name: Car Total: Payment:	Saving Goal not added	Saving Goal not added	Pass
16			Add goal with negative amounts		Goal Name: Car Total: -1123 Payment: -23	Saving Goal not added	Saving Goal added	Fail
17			Add goal with valid fields		Goal Name: Car Total: 1123 Payment: 23	Saving Goal added	Saving Goal added	Pass
18	Validation of ability of user to set a appropriate budget.	1. server is running 2. Login to preexisting user account	Set a positive integer as the budget	1. Launch App 2. Login 3. go to add page 4. Click add budget 5. insert value	Budget Amount: 50	Budget: \$50	Budget: \$50	Pass
19			Set a floating point value as the budget		Budget Amount: 0.111111	Budget: \$0.11	Budget: \$0.11	Pass
20			Set a string as the budget		Budget Amount: ABC	Cannot enter string	Cannot enter string	Pass
21			Set a negative value as the budget		Budget Amount: -50	Cannot enter negative	Budget: \$-50	Fail
22			Can change budget		Previous Budget: \$50 Changed Budget: \$20	Budget: \$20	Budget: \$50	Fail
23	Validation of ability of user to manipulate income items.	1. server is running 2. Login to preexisting user account	Set a positive integer as the income	1. Launch App 2. Login 3. go to add page 4. click add budget / edit budget / delete budget 5. Input appropriate values	Transaction Type: Income Category: salary Name: Prog Value: 22	Value: \$22	Value: \$22	Pass
24			Set a string as the income		Transaction Type: Income Category: salary Name: Prog Value: aba	Reject input	Value: \$aba	Fail
25			Set a negative value as the income		Transaction Type: Income Category: salary Name: Prog Value: -22	Reject input	Value: \$-22	Fail
26			Delete an income		Remove income with Value: \$22	Remove income	Remove income	Pass
27			Edit an income with a positive integer		Previous Income: \$22 Changed Income: \$1	Value: \$1	Value: \$1	Pass

28			Leave fields empty		Transaction Type: Category: Name: Value:	Reject input	Transaction Type: Category: food Name: Value: \$	Fail
29	Validation of ability of user to manipulate expense items.	1. server is running 2. Login to preexisting user account	Set a positive integer as the expense	1. Launch App 2. Login 3. go to add page 4. manipulate expenses	Transaction Type: Expense Category: Housing Name: Prog Value: 22	Value: \$22	Value: \$22	Pass
30			Set a string as the expense		Transaction Type: Expense Category: Housing Name: Prog Value: aba	Reject input	Value: \$aba	Fail
31			Set a negative value as the expense		Transaction Type: Expense Category: Housing Name: Prog Value: -22	Reject input	Value: \$-22	Fail
32			delete an expense		Remove expense with Value: \$22	Remove income	Remove income	Pass
33			edit an expense with a positive integer		Previous Expense: \$22 Changed Expense: \$1	Value: \$1	Value: \$1	Pass
34			Leave fields empty		Transaction Type: Category: Name: Value:	Reject input	Transaction Type: Category: food Name: Value: \$	Fail
35	Validation of user ability to view information on the dashboard	1. server is running 2. Login to account or register	View cash flow for the given month	1. Launch App 2. Login / Register	Total Income : 12,000 Total Expense: 5,000	Cash Flow: 7000	Cash Flow: 7000	Pass
36			View balance associated with the account based on its expenses and incomes.		Total Income : 12,000 Total Expense: 5,000 Cash Flow: 7,000	Balance: 7000	Balance: 0	Fail
37			View the income and expenses separated by month within the cartesian graph		July: 7000	2d chart with peak in 'Jul' with value 7000	2d chart with peak in 'Jul' with value 7000	Pass
38			View the expenses for the given month separated category in a pie chart		Expenses: Food: 300 Entertainment: 300	Food: 50 % of chart Entertainment: 50% of chart	Food: 50 % of chart Entertainment: 50% of chart	Pass
39			View the budget for the given month		Add Budget: 1000	Budget: 1000	Budget: 1000	Pass