

muzzammil's code

```
/* -----
```

```
    Budget Tracker Bot Logic
```

```
----- */
```

```
const STORAGE_KEY = 'btb_data_v1';
```

```
// Data model
```

```
let state = {
```

```
  transactions: [], // {id, type:'expense'|'income', amount, category, note, dateISO, recurring:
  {interval:'monthly', lastAppliedISO}}
```

```
  budgets: {}, // {category: limit}
```

```
  goal: { amount: 0 },
```

```
};
```

```
// Utilities
```

```
const uid = () => 't' + Math.random().toString(36).slice(2,9);
```

```
const nowISO = (d=new Date()) => d.toISOString();
```

```
const toDateStr = iso => new Date(iso).toLocaleString();
```

```
const formatCurrency = n => `₹${Number(n | 0).toLocaleString()}`;
```

```
// Load / Save
```

```
function loadState() {
```

```
  const raw = localStorage.getItem(STORAGE_KEY);
```

```
  if (raw) {
```

```
    try { state = JSON.parse(raw); }
```

```
    catch(e){ console.error('loadState parse',e); }
```

```
  }
```

```
}
```

```
function saveState() {
```

```
  localStorage.setItem(STORAGE_KEY, JSON.stringify(state));
```

```
}
```

```
/* --- Chat UI --- */
```

```
const chat = document.getElementById('chat');
```

```
function botSay(text, important=false) {
```

```
  const div = document.createElement('div');
```

```
  div.className = 'msg from-bot';
```

```
  if (important) div.style.border = '1px solid #f5d6d6';
```

```
  div.textContent = text;
```

```
  chat.appendChild(div);
```

```
  chat.scrollTop = chat.scrollHeight;
```

```
}
```

```
function userSay(text) {
```

```
  const div = document.createElement('div');
```

```
  div.className = 'msg from-user';
```

```
  div.textContent = text;
```

```
  chat.appendChild(div);
```

```
  chat.scrollTop = chat.scrollHeight;
```

```
}
```

```
/* --- Core functions --- */
```

```
function addTransaction(tx, save=true) {
```

```
  tx.id = tx.id || uid();
```

```
  tx.dateISO = tx.dateISO || nowISO();
```

```
  state.transactions.push(tx);
```

```
  if (save) { saveState(); renderAll(); }
```

```
  // After adding an expense, check budget alert
```

```
  if (tx.type === 'expense') checkBudgetAlert(tx.category);
```

```
}
```

```
function editTransaction(id, newData) {
```

```

const idx = state.transactions.findIndex(t=>t.id===id);

if (idx>=0) {

    state.transactions[idx] = { ...state.transactions[idx], ...newData };

    saveState(); renderAll();

    return true;

}

return false;

}

function deleteTransaction(id) {

    state.transactions = state.transactions.filter(t=>t.id!==id);

    saveState(); renderAll();

}


/* Budgets */

function setBudget(category, amount) {

    if (!category) return;

    state.budgets[category.toLowerCase()] = Number(amount);

    saveState(); renderAll();

}


/* Goals */

function setGoal(amount) {

    state.goal.amount = Number(amount);

    saveState(); renderAll();

}

function increaseGoal(byAmount) {

    state.goal.amount = Number(state.goal.amount || 0) + Number(byAmount);

    saveState(); renderAll();

}

```