

```

1  # =====
2  # Honorable Honey Badgers
3  # Personal Finance Application
4  # README File
5  # =====
6
7  ## Overview
8  The Honorable Honey Badgers Personal Finance Application is a modular
9  Python-based desktop tool for managing personal finances. It allows users
10 to register securely, track income and expenses, and generate interactive
11 reports with charts and drill-down detail.
12
13 The application is built using ttkbootstrap for a modern Tkinter GUI,
14 pandas for data management, and matplotlib for analytics and charting.
15
16 > Recommended Python Version: 3.10+ (tested on 3.11)
17
18 ## Features
19 - User Authentication
20   - Secure login and registration with password hashing (`bcrypt`)
21   - Multi-user support with isolated financial data
22
23 - Expense & Income Management
24   - Add, view, edit, and delete records
25   - Categorize and filter by type and date range
26   - Import/export financial data via CSV
27
28 - Analytics Dashboard
29   - Monthly Report - Bar chart of monthly expenses with drill-down to details
30   - Category Report - Pie chart of spending by category
31   - Top Categories Report - Ranked horizontal bar chart of highest spending
32   - Cash Flow Report - Line/stacked bar chart showing Income, Expenses, and Net
    Savings
33   - Export all charts as PNG and tabular data as CSV
34
35 - Account Management
36   - Change passwords, manage login history, and enforce password rules
37
38 ## Project Structure
39
40 ``` plaintext
41 Honorable_Honey_Badgers/
42 |__init__.py
43 |__main__.py                # Application entry point
44 |README.md                 # Technical Documentation
45 |requirements.txt          # Dependencies
46
47 |data/                     # Databases
48 |  |__finance.db           # Financial Data
49 |  |__users.db             # User Data
50
51 |Docs/                     # Documentation
52 |  |__ISTM 601 - Group 8 Python (Group Contract) - Phase 1.pdf
53 |  |__ISTM 601 - Group 8 User Manual - Phase 2.pdf
54 |  |__ISTM 601 - Group 8 Technical Manual - Phase 2.pdf
55 |  |__ISTM 601 - Group 8 User Manual - Phase 3.pdf
56 |  |__ISTM 601 - Group 8 Technical Manual - Phase 3.pdf
57
58 |gui/                      # GUI page controllers
59 |  |__init__.py
60 |  |__account_pages.py     # Account management (password, history)
61 |  |__auth.py              # User authentication (register/login)
62 |  |__budget_pages.py      # Budget overview & savings handling
63 |  |__expense_pages.py     # Expense management (CRUD, CSV import)
64 |  |__income_pages.py      # Income management (CRUD, CSV import)
65 |  |__category_pages.py    # Category management (update/delete)
66 |  |__main_app.py          # Main navigation frame

```

```

67 |
68 | analytics/                                # Reporting & visualization
69 | |   __init__.py
70 | |   category.py                          # Expenses by category (Pie chart)
71 | |   monthly.py                          # Monthly expenses (Bar chart)
72 | |   top_categories.py                   # Top categories with date filter
73 | |   cash_flow.py                       # Cash flow (Income vs Expenses line chart)
74 | |
75 | | operations/                            # Business logic & DB operations
76 | | |   __init__.py
77 | | |   database.py                       # Database setup & authentication
78 | | |   finance_operations.py             # CRUD, CSV import/export, budget summary
79 | | |   reports.py                       # Data access helpers for analytics
80 | ...
81 |
82 | ## Installation
83 |
84 | 1. **Clone or download the project files**
85 |
86 |     ```bash
87 |     git clone https://github.com/jhenderson-tamu/Honorable_Honey_Badgers.git
88 |     cd Honorable_Honey_Badgers
89 |     ```
90 |
91 | 2. **Create and activate a virtual environment**
92 |
93 |     ```bash
94 |     python -m venv .venv
95 |     source .venv/bin/activate    # On Windows use: .venv\Scripts\activate
96 |     ```
97 |
98 | 3. **Install required dependencies:**
99 |
100 |     ```bash
101 |     pip install -r requirements.txt
102 |     ```
103 |
104 | 3. **Run the application:**
105 |
106 |     - **From Command Line:**
107 |
108 |         ```bash
109 |         python main.py
110 |         ```
111 |
112 |     - **From Python IDLE:**
113 |
114 |         1. Open IDLE (Python GUI)\
115 |         2. Go to **File → Open...** and select `main.py`\
116 |         3. Click **Run → Run Module** (or press **F5**) to start the
117 |            application
118 |
119 | ## Module Descriptions
120 |
121 | ### `database.py`
122 |
123 | - Sets up `users.db` (user authentication, login history)\
124 | - Sets up `finance.db` (expenses, income, categories)\
125 | - Handles registration, authentication, and login history tracking
126 |
127 | ### `finance_operations.py`
128 |
129 | - Manages categories, expenses, and income\
130 | - CRUD operations (create, read, update, delete)\
131 | - CSV import (expenses and income)\
132 | - Budget summaries and calculations
133 |
134 | ### `reports.py`

```

```
134
135 - Provides database access for analytics\
136 - Expense and income retrieval by category or date range\
137 - Generates monthly aggregated data for charts
138
139 ### `auth.py`
140
141 - Handles user login and registration GUI\
142 - Validates credentials\
143 - Starts the main app after login
144
145 ### `expense_pages.py`
146
147 - Manual expense entry form\
148 - Expense import from CSV\
149 - View and remove expenses\
150 - Category management
151
152 ### `income_pages.py`
153
154 - Manual income entry form\
155 - Income import from CSV\
156 - View and remove income\
157 - Income category management
158
159 ### `budget_pages.py`
160
161 - Budget overview calculations\
162 - Date range selection\
163 - Displays totals, income, expenses, savings
164
165 ### `account_pages.py`
166
167 - Change password interface\
168 - View login history\
169 - Account management options
170
171 ### `category_pages.py`
172
173 - Manage categories for both expenses and income\
174 - Update, rename, or delete with reassignment handling
175
176 ### `main_app.py`
177
178 - Builds main window after login\
179 - Sidebar navigation for all modules\
180 - Handles logout and page switching
181
182 ### `analytics/`
183
184 - **category.py**: Expenses by category pie chart\
185 - **monthly.py**: Monthly expenses bar chart\
186 - **top_categories.py**: Top categories ranked by amount\
187 - **cash_flow.py**: Cash flow line chart (Income vs Expenses vs Net)
188
189 ### `main.py`
190
191 - Initializes databases\
192 - Starts authentication\
193 - Launches main app
194
195 ## Usage
196
197 ### First Time Setup
198
199 1. Run the app with `python main.py`\
200 2. Register a new user\
```

```
201 3. Login to access the main menu
202
203 ### Expenses & Income
204
205 - Add manually or import via CSV\
206 - View or delete existing records\
207 - Manage categories
208
209 ### Budget Overview
210
211 - Choose a start and end date\
212 - View totals, income, expenses, and net savings
213
214 ### Analytics
215
216 - Pie chart of expenses by category\
217 - Bar chart of monthly expenses\
218 - Bar chart of top categories\
219 - Line chart for cash flow
220
221 ### CSV Import Format
222
223 Both **expense** and **income** CSV files must include:\
224 - `date` (YYYY-MM-DD)\
225 - `category` (string)\
226 - `amount` (numeric)\
227 - `description` (optional)
228
229 ## Database Files
230
231 - `users.db`: Accounts and login history\
232 - `finance.db`: Expenses, income, categories
233
234 ## Security Features
235
236 - User passwords are hashed using bcrypt before storage
237 - No plaintext credentials are saved in the database
238 - Isolated user data\
239 - Session-based authentication
240
241 ## Dependencies
242
243 - **pandas**: Data manipulation\
244 - **bcrypt**: Password hashing\
245 - **tkbootstrap**: GUI framework\
246 - **matplotlib**: Charts & visualizations
247
248 ## Troubleshooting
249
250 1. **Missing imports**: Install dependencies with
251   `pip install -r requirements.txt`\
252 2. **Database errors**: Delete `.db` files to reset\
253 3. **GUI issues**: Ensure `tkbootstrap` is installed
254
255 ## Future Enhancements
256
257 - Recurring transactions\
258 - Search and filtering\
259 - Data export\
260 - Backup/restore
261
```