# Cocktail Mixing System Documentation

## Table of Contents

1. [System Overview](#4spjtj29q48p)
2. [Prerequisites](#p2yppan7ppfa)
3. [Installation](#s8bm8afytr9e)
4. [Raspberry Pi Setup](#iedaq9xweoja)
5. [Windows Setup](#w4hch49e6bmf)
6. [Application Usage](#32e9xawdyvf)
7. [Troubleshooting](#7i2jsrmxw7bk)
8. [Maintenance](#uabtpf5u84rs)

## System Overview

The Cocktail Mixing System is an automated cocktail preparation system that combines:

* A web-based user interface for cocktail selection and management
* A Python backend for processing orders and controlling hardware
* Serial communication with Arduino for pump control
* Firebase integration for data synchronization
* Support for both Windows and Raspberry Pi platforms

Main Interface *The main interface of the Cocktail Mixing System, showing the cocktail selection menu and system status.*

## Prerequisites

### Common Requirements

* Python 3.11 or later
* Node.js and npm
* Git
* Firebase account and credentials
* Arduino with pump control setup

### Raspberry Pi Specific

* Raspberry Pi 4 (recommended) or Raspberry Pi 3
* MicroSD card (16GB or larger)
* Power supply
* HDMI cable and monitor (for initial setup)
* Keyboard and mouse (for initial setup)
* Internet connection

## Installation

The installation process is automated using the installer.sh script, which handles all the necessary setup steps. Simply run:

*# Run the installer*  
./installer.sh

The installer script will:

* Install all required Python dependencies
* Set up Node.js and Electron
* Configure Firebase
* Set up the application for your platform (Windows or Raspberry Pi)
* Configure auto-start settings

For manual installation or troubleshooting, refer to the detailed steps in the troubleshooting section.

## Raspberry Pi Setup

### 1. Initial Setup

*# Update system*  
sudo apt update  
sudo apt upgrade  
  
*# Install required packages*  
sudo apt install python3-pip python3-venv git nodejs npm tmux cron

### 2. Configure Serial Port

sudo raspi-config

* Navigate to "Interface Options"
* Enable Serial Port
* Disable Serial Console
* Reboot when prompted

### 3. Install Application

*# Clone repository*  
git clone https://github.com/DeamonKing/new-repo.git /opt/cocktail-mixer  
cd /opt/cocktail-mixer  
  
*# Install dependencies*  
pip3 install -r requirements.txt  
npm install -g electron

### 4. Configure Auto-Start

The installer will automatically set up crontab for startup. To verify:

crontab -l

Should show:

@reboot /opt/cocktail-mixer/start\_at\_boot.sh

## Windows Setup

### 1. Install Required Software

* Install Python 3.11 from [python.org](https://www.python.org/downloads/)
* Install Node.js from [nodejs.org](https://nodejs.org/)
* Install Git from [git-scm.com](https://git-scm.com/)

### 2. Install Application

*# Clone repository*  
git clone https://github.com/DeamonKing/new-repo.git  
cd new-repo  
  
*# Install dependencies*  
pip install -r requirements.txt  
npm install -g electron

### 3. Configure Auto-Start

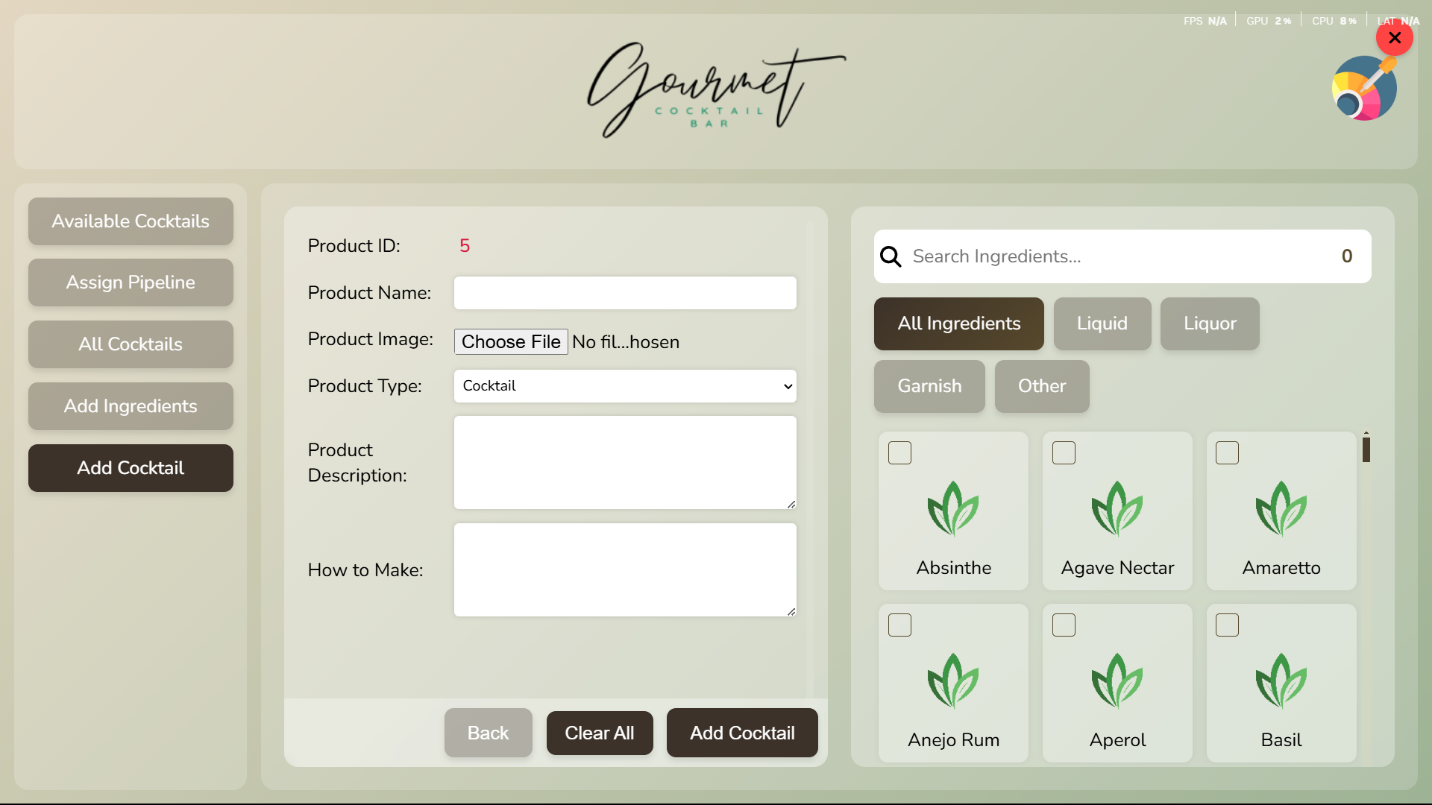
1. Create a shortcut to start\_at\_boot.sh
2. Place the shortcut in the Startup folder:
   * Press Win+R
   * Type shell:startup
   * Copy the shortcut to this folder

## Application Usage

### Managing Cocktails

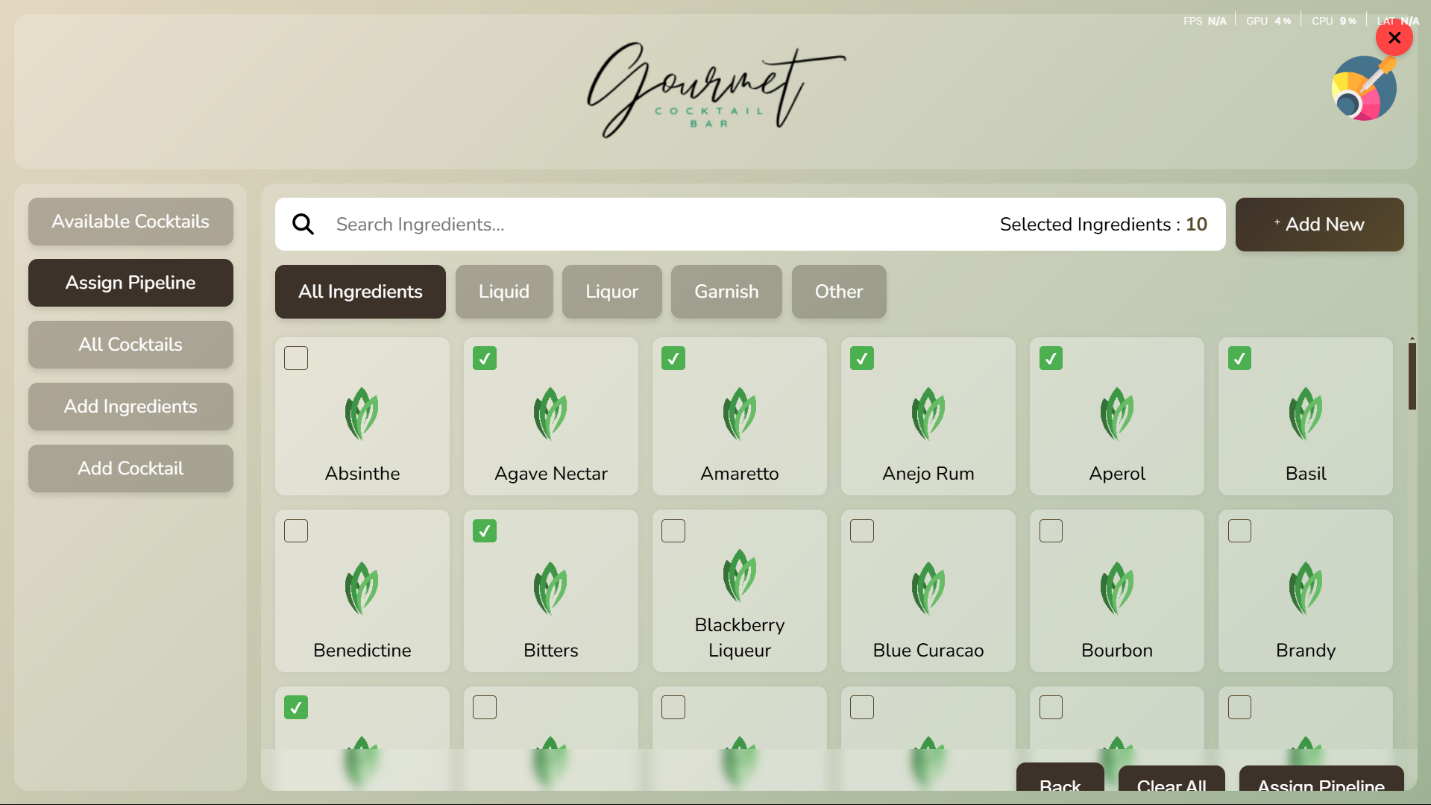
Cocktail Management *The cocktail management screen where users can add, edit, and delete cocktails from the system.*

1. **Add New Cocktail**



* + Click "Add Cocktail"
  + Enter cocktail details
  + Assign ingredients to cocktail
  + Set drink intensity
  + Save the cocktail

### Managing Ingredients



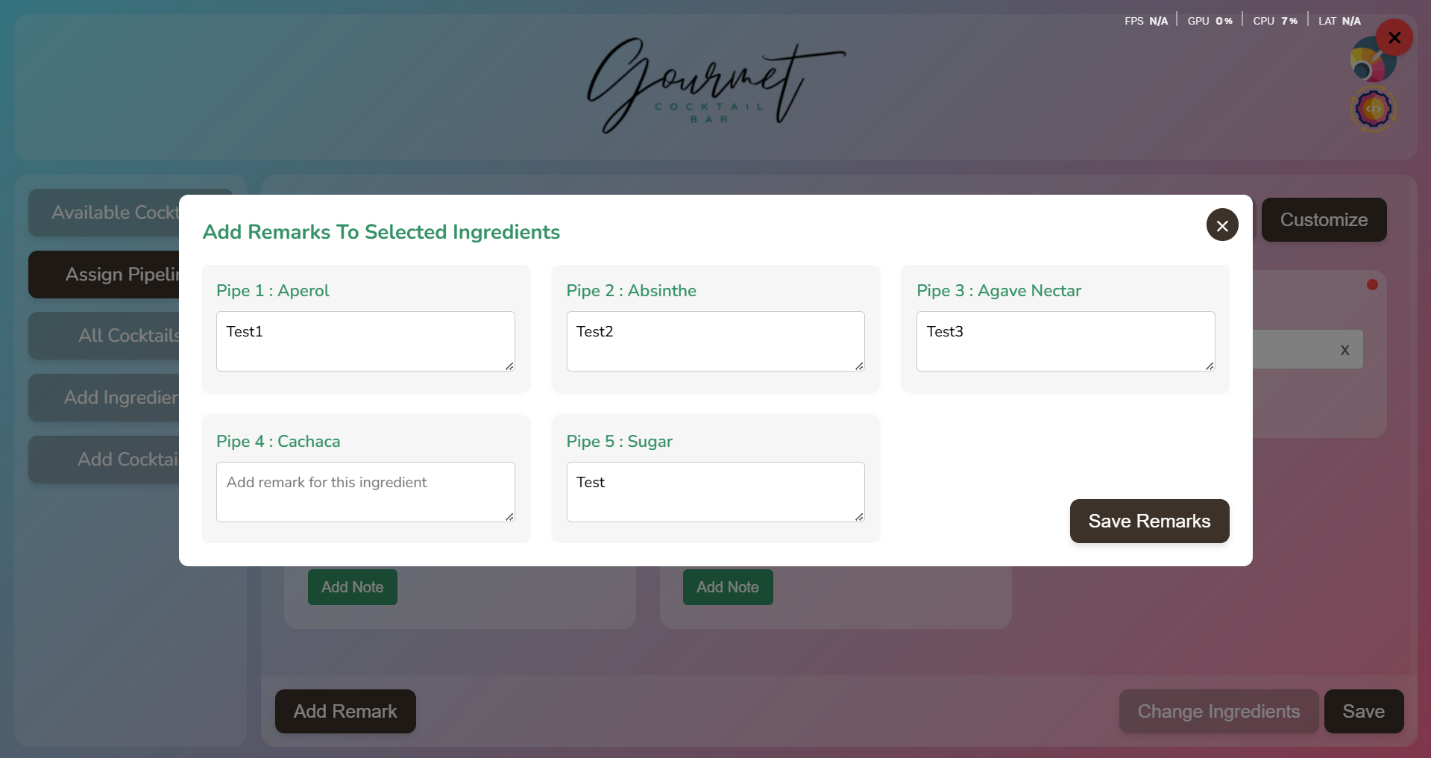
Ingredient Configuration *The ingredient configuration interface for managing available ingredients and their pump assignments.*

1. **Add New Ingredient**



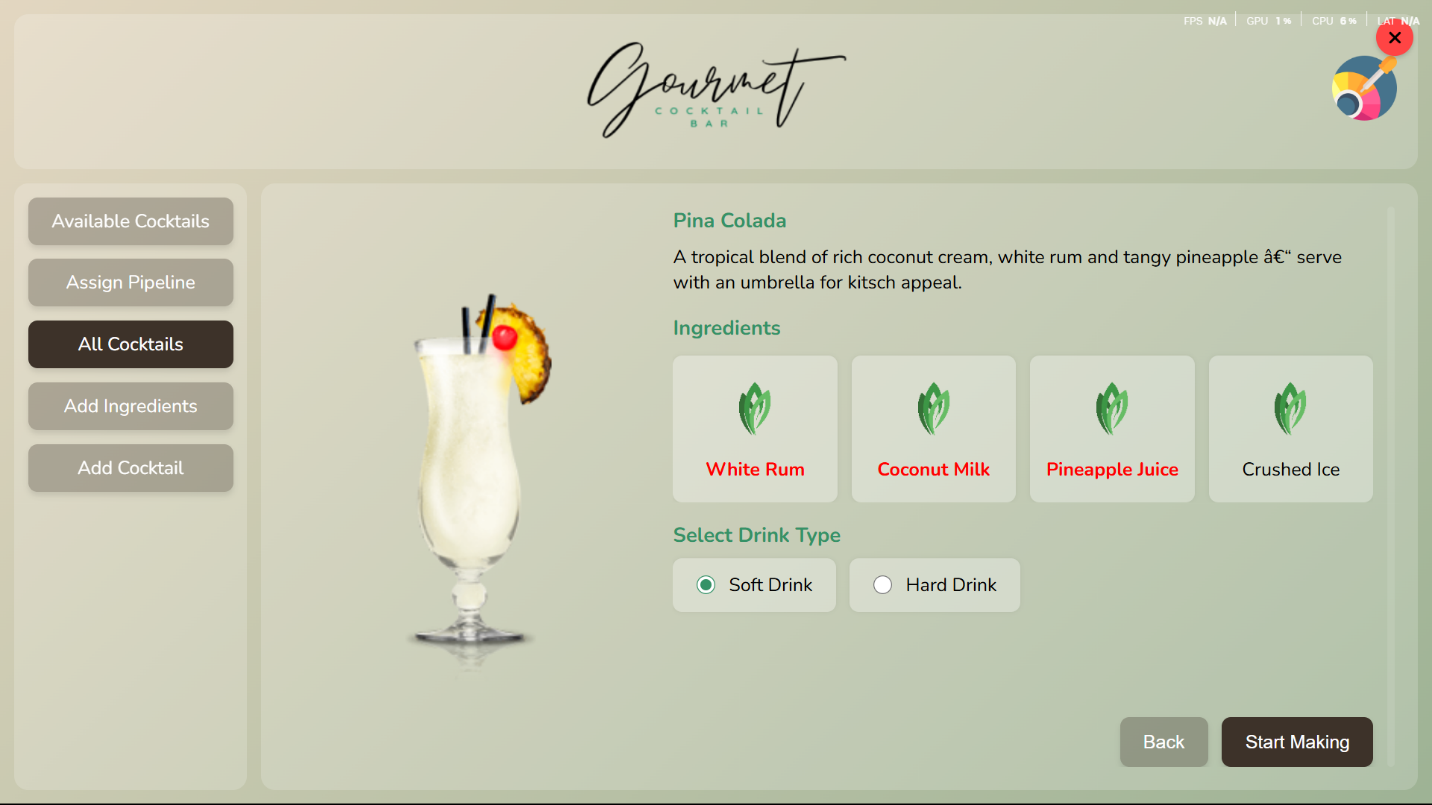
* + Click "Add Ingredient"
  + Enter ingredient details
  + Assign to a pump
  + Save the ingredient

1. **Update Ingredients remarks**



* + Select ingredients to modify
  + Update quantities or pump assignments
  + Save changes

### Making Cocktails



Drink Intensity *The drink intensity selection screen where users can choose between light, medium, and strong drink options.*

1. Select a cocktail from the menu
2. Choose drink intensity (light/medium/strong)
3. Click "Make Cocktail"
4. The system will automatically:
   * Control pumps for precise measurements
   * Monitor the mixing process
   * Display progress
   * Notify when complete

Mixing Process *The mixing process screen showing real-time progress of cocktail preparation.*

### Update APP

*  
  
*

* Check for newer versions of the application.
* Download and install the latest version with a simple click of the "Update" button.
* Ensure the system is running the most up-to-date software.

## Configuration and Credential Files

* jecon-cocktail-machine-firebase-adminsdk-fbsvc-0fecbeb26b.json - Firebase Admin SDK credentials file for authentication
* firebase\_config.py - Contains Firebase configuration settings and initialization
* firebase\_credentials\_encoded.txt - Encoded Firebase credentials for secure storage
* config.py - General application configuration settings
* requirements.txt - Python package dependencies list
* package.json - Node.js package dependencies and scripts

## Main Application Files

* app.py - Main application file containing the core functionality
* image\_handler.py - Handles image processing and management
* loading\_screen.py - Manages the loading screen functionality

## Git and Version Control

* .git/ - Git repository directory
* .gitignore - Specifies which files Git should ignore
* git-pull.py - Script for pulling updates from Git repository
* git-push.py - Script for pushing changes to Git repository

## System and Startup Scripts

* run.sh - Main script to run the application
* start\_at\_boot.sh - Script to start the application at system boot
* keyboardstart.sh - Script to start keyboard-related functionality
* keyboardstop.sh - Script to stop keyboard-related functionality

## Utility Scripts

* encode\_credentials.py - Script for encoding credentials securely

## Directories

* static/ - Contains static files like images, CSS, and JavaScript
* node\_modules/ - Contains Node.js dependencies
* \_\_pycache\_\_/ - Python bytecode cache directory
* backups/ - Directory for storing backup files
* .vscode/ - Visual Studio Code configuration directory
* Jecon\_CM\_installer/ - Directory containing installer files

## Static Directory Contents

### Frontend Files

* index.html - Main HTML file for the web interface
* style.css - CSS styles for the application
* script.js - Main JavaScript file for frontend functionality
* colorPicker.js - JavaScript for color picker functionality
* main.js - Main JavaScript entry point
* preload.js - Preload script for Electron app

### Data Files

* config.json - Configuration data for the application
* products.json - Product information and data
* db.json - Database file containing application data

### Processing Files

* processing - File indicating processing status
* processing\_complete - File indicating processing completion

### Images Directory

* img/ - Directory containing image assets

## Project Overview

This is a cocktail machine application with both backend (Python) and frontend (JavaScript/HTML/CSS) components. It uses Firebase for backend services and data storage. The application includes features for managing products, handling images, and providing a user interface for cocktail machine operations.

## Troubleshooting

## Common Issues

1. **Application Won't Start**
   * Check if port 5000 is available
   * Verify Python and Node.js installations
   * Check system logs
2. **Serial Communication Problems**
   * Verify Arduino connection
   * Check serial port configuration
   * Review system logs
3. **Firebase Sync Issues**
   * Verify internet connection
   * Check Firebase credentials
   * Review error logs
4. **Pump Control Problems**
   * Check serial connection
   * Verify pump assignments
   * Review error logs

## Support

For additional support:

* Check the [GitHub repository](https://github.com/DeamonKing/new-repo)
* Review the troubleshooting section
* Contact system administrator