

Task Scheduler - PHP Implementation

This is a complete implementation of the PHP-based task management system with email subscription functionality.

Files Overview

Core Files

- `functions.php` - Contains all required functions for task and email management
- `index.php` - Main user interface for task management and email subscription
- `verify.php` - Handles email verification when users click verification links
- `unsubscribe.php` - Handles unsubscribe requests from email links
- `cron.php` - Script that runs hourly to send task reminder emails
- `setup_cron.sh` - Bash script to automatically configure the CRON job

Data Files (JSON format)

- `tasks.txt` - Stores all tasks in JSON format
- `subscribers.txt` - Stores verified email subscribers
- `pending_subscriptions.txt` - Stores pending email verifications with codes
- `cron_log.txt` - Log file for CRON job execution (created automatically)

Setup Instructions

1. **Upload all files** to your web server in the `src/` directory

2. **Set proper permissions:**

```
bash
```

```
chmod 755 setup_cron.sh
```

```
chmod 644 *.php *.txt
```

3. **Run the CRON setup script:**

```
bash
```

```
./setup_cron.sh
```

4. **Configure your server** to send emails (ensure PHP `mail()` function works)

Features Implemented

✓ Task Management

- Add new tasks (prevents duplicates)
- Mark tasks as complete/incomplete
- Delete tasks
- All tasks stored in JSON format in `tasks.txt`

✓ Email Subscription System

- Users can subscribe with email address
- 6-digit verification code generation
- Email verification process with clickable links
- Verification emails sent from `no-reply@example.com`
- Pending subscriptions stored until verified
- Verified subscribers moved to main subscriber list

✓ **Reminder System**

- CRON job runs every hour (configured by `setup_cron.sh`)
- Sends HTML-formatted emails to all verified subscribers
- Only includes pending (incomplete) tasks in reminders
- Includes unsubscribe link in every email
- One-click unsubscribe functionality

✓ **Email Content**

All emails follow the exact format specified:

Verification Email:

- Subject: "Verify subscription to Task Planner"
- Contains verification link with proper ID attribute
- Sent from: no-reply@example.com

Task Reminder Email:

- Subject: "Task Planner - Pending Tasks Reminder"
- HTML formatted with task list
- Includes unsubscribe link with proper ID attribute

Technical Implementation Details

Data Storage Format

All data follows the exact JSON schemas required:

Tasks (`tasks.txt`):

json

```
[
  {
    "id": "unique_task_id",
    "name": "Task Name",
    "completed": false
  }
]
```

Subscribers (`subscribers.txt`):

json

```
["user1@example.com", "user2@example.com"]
```

Pending Subscriptions (`pending_subscriptions.txt`):

json

```
{
  "user1@example.com": {
    "code": "123456",
    "timestamp": 1717694230
  }
}
```

Form Elements

All form elements use the exact naming and ID attributes specified:

- Task input: `name="task-name"` and `id="task-name"`
- Add task button: `id="add-task"`
- Email input: `name="email"`
- Submit email button: `id="submit-email"`
- Task list: `class="tasks-list"`
- Task items: `class="task-item"` (with `completed` class when done)
- Task checkboxes: `class="task-status"`
- Delete buttons: `class="delete-task"`

Security Features

- Email validation and sanitization
- XSS protection with `htmlspecialchars()`
- Duplicate task prevention
- Proper URL encoding for email parameters
- Case-insensitive email handling

CRON Job Details

The `setup_cron.sh` script automatically configures a CRON job that:

- Runs every hour (`0 * * * *`)
- Executes `cron.php` using PHP CLI
- Logs all activity to `cron_log.txt`
- Handles errors gracefully

Testing the System

1. **Add Tasks:** Use the main interface to add tasks
2. **Subscribe:** Enter an email address to receive verification email
3. **Verify:** Click the verification link in the email
4. **Check Reminders:** Wait for the hourly CRON job or run `php cron.php` manually
5. **Unsubscribe:** Click unsubscribe link in any reminder email

Troubleshooting

- **Emails not sending:** Check PHP `mail()` configuration on your server
- **CRON not working:** Verify CRON service is running and permissions are correct
- **Permission errors:** Ensure web server can read/write to data files
- **JSON errors:** Check that data files contain valid JSON format

Compliance Checklist

- ✓ No function names changed
- ✓ No files modified outside src/ directory
- ✓ No database used - only text files
- ✓ All required functions implemented
- ✓ Proper HTML email formatting
- ✓ Working CRON job implementation
- ✓ Exact form element naming
- ✓ JSON data storage format
- ✓ Email verification and unsubscribe
- ✓ Duplicate task prevention