**An Installation and Operation Guide**

Jeffrey Muffett

CSC425,

Professor Michael Greco,

Trident University International

March 16, 2023

**An Installation and Operation Guide**

Mendes (2022) claims that creating a web application is one of the most challenging and rewarding jobs an entrepreneur can take on. A business can increase and gain a competitive edge over competitors in a particular field if it creates a robust, client-centered web app. This document offers a step-by-step manual for setting up and using a time-tracking application. The product will be well-tested and have all the necessary functionality. The application's suitability for release will be determined by the outcomes of the test plan created in the Case Study.

# **Summary**

The Timetracker app has been fully developed and is now ready for deployment. There are two options for using the web app:

1. Deploy the war folder attached. GWT compiles java into javascript that can be rendered in a web browser. The war folder attached contains the compiled project and can be deployed with the [Deploying Time tracker Web App](#_Deploying_Time_tracker) instructions.
2. Compiling the project and deploying it to a platform of choice. In this option, you must compile the project and deploy it afterward. Follow the [System Requirements](#_System_Requirements) and [Installation Steps](#_Installation_Steps).

# **Deploying Time tracker Web App**

Now our application is working fine, and we are ready to export it as a war file.

Follow the following steps −

* Go into your project's war directory C:\<path>/ \timetracker\war
* Select all the files & folders available inside the war directory.
* Zip all the selected files & folders in a file called HelloWorld.zip.
* Rename timetracker.zip to timetracker.war.

**Deploy WAR file**

* Copy the timetracker.war file to tomcat installation directory > webapps folder.
* Start the tomcat server.
* Look inside webapps directory. There should be a folder timetracker got created.

# **System Requirements**

The system should have functionalities that support user participation and interaction with the program (Vora, 2009). Before installing the employee tracking web app, make sure your system meets the following requirements:

1. Operating System: Windows, Linux, or macOS
2. Java Development Kit (JDK) 8 or higher
3. Web servers such as Apache Tomcat
4. IDE such as Eclipse, NetBeans, or IntelliJ IDEA
5. Database management systems such as MySQL, PostgreSQL, or Oracle
6. Web browsers such as Google Chrome, Mozilla Firefox, or Safari

# **Installation Steps**

1. [Download GWT Plugin FOR Eclipse](https://www.gwtproject.org/download.html) or install a plugin from the eclipse marketplace.
2. Connect the web app to the database by setting up a database connection in your IDE.
3. Open the timetracker web application project in your IDE.
4. Import the project into Eclipse. In Eclipse, go to the File menu and choose:
5. File -> Import... -> Existing Projects into Workspace. Browse to the directory containing this file, and select "Timetracker ."Be sure to uncheck "Copy projects into workspace" if it is checked. Click Finish.
6. To build the project, left-click on "Timetracker" and choose GWT compile. After the successful build, run your project by left-clicking on “Timetracker” and choosing run as gwt project superdev mode.
7. To launch your web app in GWT in a browser, left-click on the URL in the terminal and choose the browser of your choice. Run tests and proceed to [Deploying Time tracker Web App](#_Deploying_Time_tracker).

# **Project map**

+---.settings

+---gwt-unitCache

+---src

| \---com

| \---timetracker

| | Timetracker.gwt.xml

| +---client

| +---server

\---shared

+---test

+---test-classes

\---war |

+---images

+---timetracker

\---WEB-INF

+---classes

+---deploy

\---lib

.settings: Contains all eclipse, GWT plugin settings and preferences

gwt-unitCache: Contains GWT cache generated from running the program

src: Contains server, client, shared, and Timetracker.gwt.xml

* Timetracker.gwt.xml- Contains XML code used to describe data and encode the data into plain text.
* Client - contains all client-side code with java classes to perform operations.
* Server – Contains all backend code and data storing employee, employer, and task data in SQL format.
* Shared- contains code that both client and server-side share.

Test: Contains all data generated during testing.

War: Contains CSS, HTML, icons, compiled CSS and javascript cache generated while compiling the program in dev and superdev mode, web inf, deploy, and lib containing GWT servlet. After compiling the project, the contents of this war folder can be deployed to Google App Engine or Google Cloud Platform to make the web app live.

**Changes made since the last submission**

1. The following java classes were added:

* CheckLoginServlet.java: Deals with log in. Checks if the employee/employer password and username is in the database and if they are correct.
* DeleteServelet.java: Deals with the deletion of employees and tasks.
* EditRecordServlet.java: is used to update task and employee details.
* EmployeeTable.java:
* MainView.java: handles how every operation is displayed. Calls other views.
* MenuView.java: Contains all possible operations. Serves the same purpose as a navigation bar
* RegisterServlet.java: handles user registration. Collects user details and sends them to the server side.
* ShowEmployeeServlet.java: handles how employees are displayed. Extracts employee data from the server side
* ShowTasks.java: handles how tasks are displayed. Extracts task data from the server side

1. Employee.sql and tasks.sql were added in the server folder to store employee and task data respectively
2. Compiled bootstrap was added in the ./war/css folder to handle styling. This was to reduce the code needed hence decreasing loading time and chances of code-breaking.
3. The images folder was added to the ./timetracker/war folder. This folder contains the logo and icon for the project.
4. Files in the /com/timetracker/shared folder were removed since their functions were taken over by java servlet classes added on the client side.

**How to use**

The program is meant to be used by both employees and employers. It can be used in two ways: as a web app (no installation required) and as a desktop application. The web application uses web technologies and falls under the category of client-side, with JavaScript being the principal programming language (Brewster, 2023).

# **Proof of Concept**

Our proof-of-concept software for a time-tracking system showed that the suggested method is workable and efficient. From the onset, the system was designed to help businesses in effective time management and following up on staff activities and projects. As part of our proof of concept, we created a software prototype including elements like task creation and assignment, progress monitoring, and reporting—the program's performance and scalability under various conditions, including user load and data volume conditions. Our proof-of-concept results were promising; they showed that the program could effectively manage and track employee projects and activities while providing management with valuable data and reporting capabilities.

**STATUS**

Users who use incompatible browser versions run into several difficulties. These difficulties highlight the need for good browsers to run effectively and efficiently for the program. Some older browser versions might need to be completely compatible with the technology since the APP was built using Google Web Toolkit (GWT). Another significant issue is the need for more tools and documentation for GWT, making it difficult for us to troubleshoot and fix potential problems. Browser compatibility checks have been implemented to ensure users are using compatible browser versions to resolve some of these issues. A knowledge library and support system have also been developed to assist users in troubleshooting any issues they might run into while utilizing the app. Additionally, we keep an eye on the app and adjust to better its compatibility and stability while exploring other technologies that may be more user-friendly and widely supported.

# **References**

Brewster, C. (n.d.). Web app development in 2023: Everything you need to know - Trio. Trio Developers - Stop searching. Start building. <https://www.trio.dev/blog/web-app-development>

Mendes, A. (2022, December 8). *Web app development: The ultimate guide for 2023*. Blog | Imaginary Cloud. <https://www.imaginarycloud.com/blog/web-app-development-the-ultimate-guide/>

Vora, P. (2009). Rich Internet Applications. Just a moment... <https://www.sciencedirect.com/topics/computer-science/traditional-web-application>