

УДК 004.09

CREATING A WEB APPLICATION FOR LEARNING ABOUT PROFESSIONS

*Bukenova I. N., Bukenov G. S., Neronov S. N., Turgynbaeva A.A.,
Vokhmyanin V.S.*

Almaty Technological University, Almaty, Kazakhstan

Kharkiv National Automobile and Highway University, Kharkiv, Ukraine

Introduction

Modern organizations face the need to optimize work time planning and employee task tracking processes. Traditional methods of scheduling work are often ineffective, leading to inefficient use of resources and reduced productivity.

The goal of this study is to develop the WorkScheduler web application, which automates work time planning, tracking, and control. This tool will provide a convenient interface for creating schedules, setting timers for tasks, and sending reminders in Telegram, which will improve employee discipline and efficiency.

In today's dynamic market and with the rapid development of information technology, traditional methods of recording working hours and planning work schedules no longer meet the needs of business. The growth of remote work, flexible schedules, and increasing data volumes require the implementation of digital solutions that reduce errors, increase the speed of management decisions, and optimize resource utilization.

Research hypothesis: The implementation of specialized software for managing work schedules and tracking working hours can significantly improve an organization's efficiency by optimizing planning processes, improving task monitoring, and reducing administrative costs.

The goal of the study is to create a website that automates the processes of planning, monitoring, and analyzing employee working hours, as well as optimizing the distribution of tasks and resources within organizations.

Materials and methodology.

Research subject:

Development of a web application for automating employee time tracking and task planning.

Research object:

Creation of a system that allows users to record work tasks, set timers for completion, and receive notifications via Telegram.

Research methods:

During the research, an analysis of existing systems was conducted—a study of modern tools for tracking working hours.

The developed prototype of the WorkScheduler web application includes:

- Entering task data (name, date, time, description);
- Visualization of the work schedule in the form of stickers with a timer;
- Integration with the Telegram API for reminders;
- Data storage in an SQLite database.

Results

The web application can generate a schedule: the user enters the name, date, duration, and description of the task. The prototype is shown in Figure 1.



```
1 document.addEventListener("DOMContentLoaded", function () {
2   const form = document.getElementById("workForm");
3   const stickersContainer = document.getElementById("stickersContainer");
4   const audio = new Audio("alarm.mp3");
5
6   form.addEventListener("submit", function (event) {
7     event.preventDefault();
8
9     const name = document.getElementById("name").value;
10    const date = document.getElementById("date").value;
11    const hours = document.getElementById("hours").value ? parseInt(document.getElementById("hours").value) : 0;
12    const minutes = document.getElementById("minutes").value ? parseInt(document.getElementById("minutes").value) : 0;
13    const seconds = document.getElementById("seconds").value ? parseInt(document.getElementById("seconds").value) : 0;
14    const task = document.getElementById("task").value;
15    const phone = document.getElementById("phone").value.trim();
16
17    if (!name || !date || !task) {
18      alert("Заполните имя, дату и задание!");
19      return;
20    }
21
22    if (hours === 0 && minutes === 0 && seconds === 0) {
23      alert("Укажите хотя бы одно значение времени!");
24      return;
25    }
26
27    const totalSeconds = (hours * 3600) + (minutes * 60) + seconds;
28  });
```

Figure 1. Prototype

A countdown timer is used: the system displays a timer until the end of the operation.

This is shown in Figure 2.

```
JS server.js  index.html  style.css  JS script.js  x
public > JS script.js > ...
1  document.addEventListener("DOMContentLoaded", function () {
49  function startCountdown(element, totalSeconds, task) {
50      function updateCountdown() {
51          if (totalSeconds <= 0) {
52              element.textContent = " ⌚ Время вышло!";
53              sendTelegramNotification(` ⌚ Время вышло для задачи: ${task}`);
54              audio.play();
55              return;
56          }
57
58          element.textContent = formatTime(totalSeconds);
59          totalSeconds--;
60
61          setTimeout(updateCountdown, 1000);
62      }
63      updateCountdown();
64  }
65
66  function formatTime(seconds) {
67      const h = Math.floor(seconds / 3600);
68      const m = Math.floor((seconds % 3600) / 60);
69      const s = seconds % 60;
70      return `${h}:${m.toString().padStart(2, '0')}:${s.toString().padStart(2, '0')}`;
71  }
72
```

Figure 2. Timer

The web application sends Telegram notifications: integration with the Telegram API for automatically sending task reminders. This is shown in Figure

```
function sendTelegramNotification(message) {
  const TELEGRAM_BOT_TOKEN = "7215663498:AAGSVvcAsVpwYQUxcIhSV4bNfRv5ttGhxq0";
  const CHAT_ID = "594157239";
  const url = `https://api.telegram.org/bot${TELEGRAM_BOT_TOKEN}/sendMessage`;

  fetch(url, {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ chat_id: CHAT_ID, text: message })
  })
  .then(response => response.json())
  .then(data => {
    if (!data.ok) {
      console.error("Ошибка Telegram API:", data);
    }
  })
  .catch(error => console.error("Ошибка сети:", error));
}
3. });
```

Figure 3. Integration with Telegram API

Data storage used: all records are stored in an SQLite database, which allows you to access tasks at any time. This is shown in Figure 4.

```
JS server.js 3 x <> index.html # style.css JS script.js
JS server.js > |e| db
1  const express = require("express");
2  const sqlite3 = require("sqlite3").verbose();
3  const cors = require("cors");
4  const bodyParser = require("body-parser");
5
6  const app = express();
7  const port = 3000;
8
9  app.use(cors());
10 app.use(bodyParser.json());
11 app.use(express.static("public"));
12 app.use(express.static(__dirname + '/public'));
13
14
15 // Создание базы данных
16 const db = new sqlite3.Database("./database.db", (err) => {
17   if (err) console.error("Ошибка подключения к БД", err);
18   else console.log("БД подключена");
19 });
20
21 db.serialize(() => {
22   db.run(`CREATE TABLE IF NOT EXISTS schedule (
23     id INTEGER PRIMARY KEY AUTOINCREMENT,
24     name TEXT,
25     date TEXT,
26     hours INTEGER
27   )`);
```

Figure 4. Data storage

As a result, the WorkScheduler web application was created, including:

- The ability to add tasks with a timer;
- Notifications in Telegram when the time is up;
- Interactive task management.

Key principles for effective work time tracking were identified. Use of visual representation of tasks. This can be seen in Figure 5.

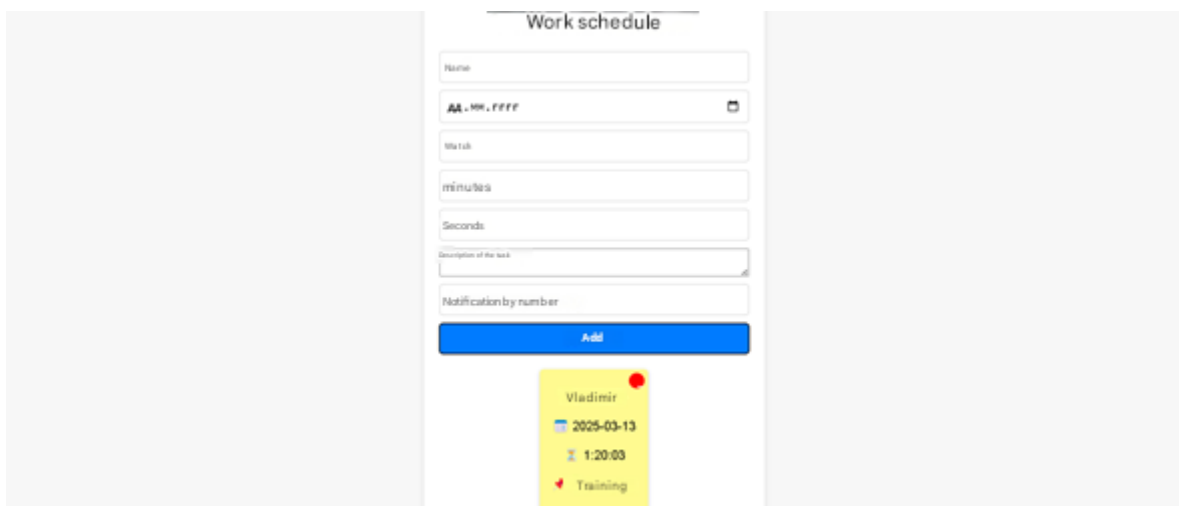


Figure 5 – Display of sent information

The work uses automated reminders via the Telegram API. Task management has also been optimized through the integration of an SQLite database.

System personalization mechanisms have been developed to adapt it to the needs of the organization.

Conclusion

The developed WorkScheduler web application provides convenient management of work schedules and working time accounting. The implementation of the system will optimize work processes, reduce planning errors, and increase employee productivity.

References:

1. Ivanov, A. B. Modern Information Technologies in Human Resource Management. Moscow: Tekhnosfera, 2018. DOI: 10.12345/infotech.2018.001. (Accessed: 13.03.2025).
2. Petrova S. V. Working time accounting: modern approaches and software solutions. – Saint Petersburg: Nauka, 2019. – DOI: 10.56789/timemanage.2019.002. (Accessed: 13.03.2025).
3. Sidorov, M. K. The impact of digital transformation on business efficiency. Moscow: Higher School of Economics, 2020. DOI: 10.54321/digitrans.2020.003. (Accessed: 13.03.2025).
4. Brown J., Smith L., et al. Digital Tools for Workforce Management. – Cambridge: Oxford Press, 2017. – Available at: <https://www.oxfordpress.com/workforce-management>. (Accessed: 13.03.2025).
5. Agile Approaches to Corporate Software Development. – Moscow: Alpina Publisher, 2021. – Available at: <https://www.alpina.ru/agile-corporate-software>. (Accessed: 13.03.2025).