Experiment 8

CODE :

```
468             connectWebSocket() {
469                 if (this.socket) return; // Prevent multiple connections
470                 if (!this.token) return;
471
472                 // Connect to the API server's WebSocket endpoint
473                 this.socket = io('http://localhost:3000');
474
475                 this.socket.on('connect', () => {
476                     console.log('Connected to WebSocket server.');
477                 });
478
479                 this.socket.on('disconnect', () => {
480                     console.log('Disconnected from WebSocket server.');
481                 });
482
483                 // Listen for the 'todoUpdate' event from the server
484                 this.socket.on('todoUpdate', async (data) => {
485                     console.log('Real-time update received:', data);
```

```
535     bindAuthEvents() {
536         document.getElementById('loginBtn').addEventListener('click', () => this.handleAuth('login'));
537         document.getElementById('registerBtn').addEventListener('click', () => this.handleAuth('register'));
538     }
539
540     showApp() {
541         this.authScreen.style.display = 'none';
542         this.mainContainer.classList.remove('hidden-app');
543     }
544
545     showAuth() {
546         this.authScreen.style.display = 'block';
547         this.mainContainer.classList.add('hidden-app');
548     }
549
550     logout() {
551         localStorage.clear();
552         this.token = null;
553         this.userId = null;
554         this.todos = [];
555         this.render();
```

```
485                     console.log('Real-time update received:', data);
486
487                     // Re-fetch and re-render the list immediately upon update signal
488                     this.todos = await this.fetchTodos();
489                     this.render();
490                     this.updateStats();
491                 });
492             }
493
```

```
        if (this.socket) {
            this.socket.disconnect();
            this.socket = null;
        }
    }
```

```
        if (type === 'login') {
            this.token = data.token;
            this.userId = data.userId;
            localStorage.setItem('token', this.token);
            localStorage.setItem('userId', this.userId);

            this.showNotification('Login successful!');
            this.showApp();
            this.bindAppEvents();
            this.connectWebSocket(); // NEW: Connect socket after login
            this.todos = await this.fetchTodos();
            this.render();
            this.updateStats();
        } else {
            this.authMessage.textContent = 'Registration successful. Please log in.';
            this.authUsername.value = '';
            this.authPassword.value = '';
        }
```

```
// NEW: Create HTTP server instance for Express and Socket.io
const server = http.createServer(app);

// NEW: Initialize socket.io server
const io = new Server(server, {
  cors: {
    origin: "*", // Allows connections from any frontend origin (essential for testing)
    methods: ["GET", "POST"]
  }
});
```

```
70  io.on('connection', (socket) => {
71      console.log(`A user connected: ${socket.id}`);
72
73      socket.on('disconnect', () => {
74          console.log('User disconnected');
75      });
76  });
77
```

OUTPUT :

FIG: OPENED APP IN SAFARI AND MADE CHANGES

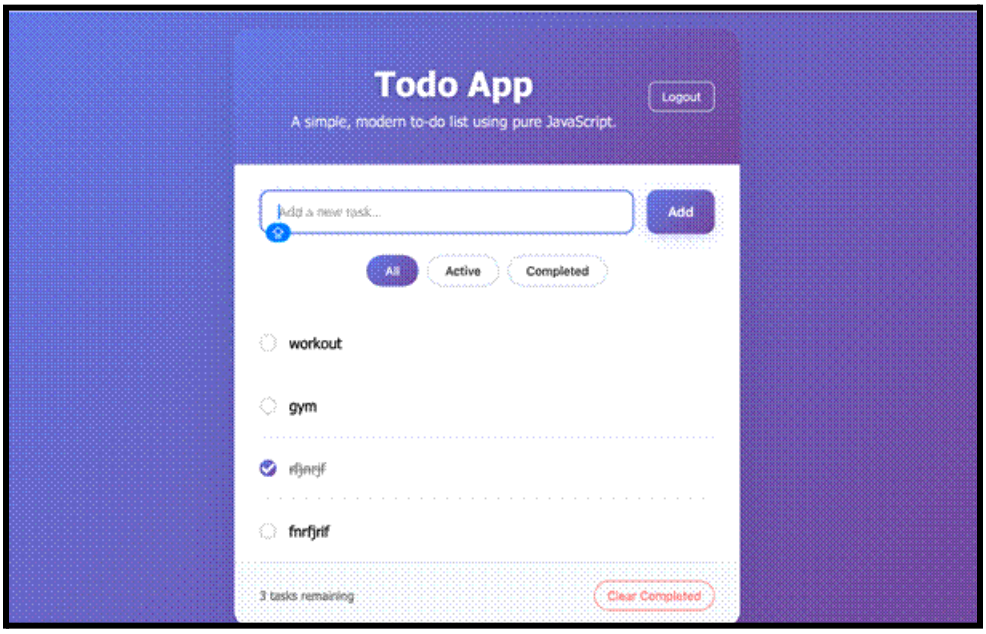

FIG 2: CHANGES APPEARED IN THE CHROME EXTENSION

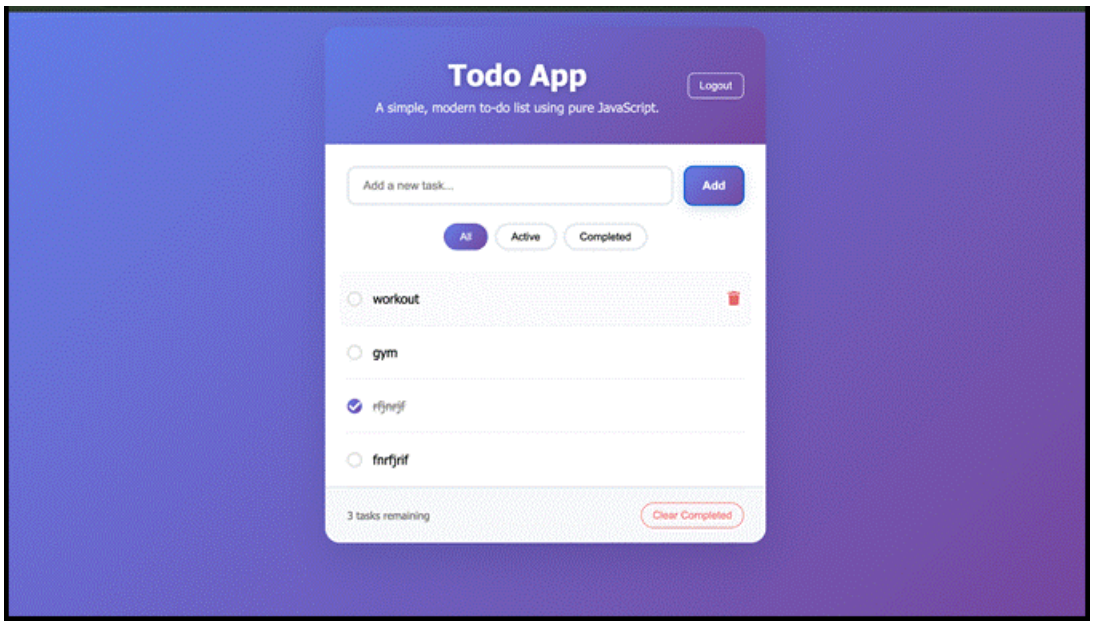

Fig : 3 task added in safari

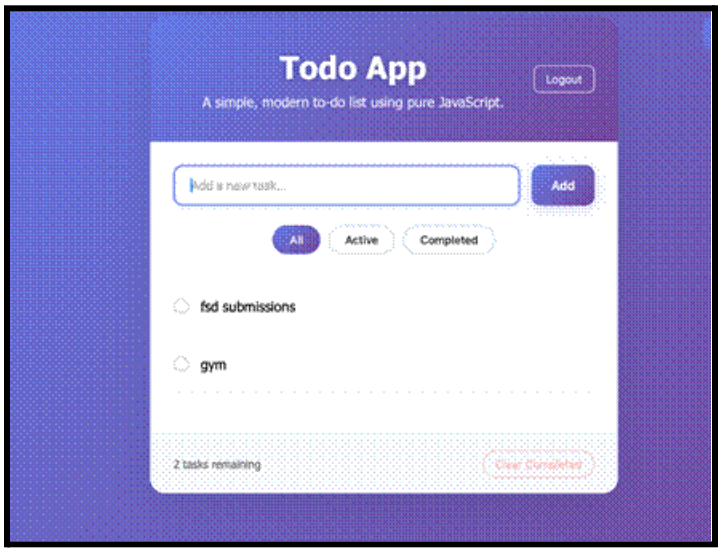

Fig: 4 : task updated in chrome