## **Kanban board for Moodle assignments**

**API Documentation**

Kanban board for Moodle assignments allows to easily manage and access assignments for both students and teachers. Thanks to the intuitive interface, users can do specified tasks in the fastest way possible, as the board has no extra or distracting details related to other activities.

**Add Subject**

**Description: allows teachers to create new subjects.**

**Request:**

**<?php**

**require(dirname(\_\_FILE\_\_) . '/includes/db.php');**

**$name = $\_POST["name"];**

**$sql = "INSERT INTO subjects (name) VALUES ('$name')";**

**if ($conn->query($sql) === TRUE) {**

**echo json\_encode(["message" => "success"]);**

**} else {**

**echo json\_encode(["error" => "Error: " . $sql . "<br>" . $conn->error]);**

**}**

**header("Location: /teacher");**

**die();**

**Add task**

**Description: allows teachers to create new tasks.**

**Request:**

**<?php**

**require(dirname(\_\_FILE\_\_) . '/includes/db.php');**

**$subject\_id = $\_POST['subject\_id'];**

**$name = $\_POST['name'];**

**$description = $\_POST['description'];**

**$deadline = $\_POST['deadline'];**

**$status = $\_POST['status'];**

**$allowed\_statuses = ['opened', 'submitted', 'graded'];**

**if (!in\_array($status, $allowed\_statuses)) {**

**// Handle invalid status value, perhaps set a default value**

**$status = 'opened'; // or any other default value**

**}**

**$sql = "INSERT INTO assignments (subject\_id, status, name, description, deadline) VALUES ('$subject\_id', '$status', '$name', '$description', '$deadline')";**

**if ($conn->query($sql) === TRUE) {**

**echo json\_encode(["message" => "Assignment added successfully"]);**

**} else {**

**echo json\_encode(["error" => "Error: " . $sql . "<br>" . $conn->error]);**

**}**

**header("Location: /teacher");**

**die();**

**Response:**

**const CreatedTasks = () => {**

**return (**

**<div>**

**<form action='/api/add\_task.php' method='POST' class='task'>**

**<input name='name' placeholder='' />**

**<input type='date' name='deadline' id='deadline' />**

**<input name='description' placeholder='description' />**

**<input name='status' value='opened' hidden />**

**<input name='subject\_id' id='subject\_id' hidden value={subjectId} />**

**<button type='submit'>Add task</button>**

**</form>**

**{tasks.filter((t) => t.status === 'opened').map((task) => (**

**<div class='task'>**

**<h3>{task.name}</h3>**

**<div class='desc'>{task.description}</div>**

**<time>{task.deadline}</time>**

**</div>**

**))}**

**</div>**

**)**

**}**

**Get subjects**

**<?php**

**require(dirname(\_\_FILE\_\_) . '/includes/db.php');**

**$sql = "SELECT \* FROM subjects";**

**$result = $conn->query($sql);**

**$subjects = [];**

**while ($row = $result->fetch\_assoc()) {**

**$subjects[] = $row;**

**}**

**echo json\_encode($subjects);**

**Get tasks**

**Description: gives students information about tasks.**

**Request:**

**<?php**

**require(dirname(\_\_FILE\_\_) . '/includes/db.php');**

**$sql = "SELECT \* FROM assignments";**

**$result = $conn->query($sql);**

**$assignments = [];**

**while ($row = $result->fetch\_assoc()) {**

**$assignments[] = $row;**

**}**

**echo json\_encode($assignments);**

**Response:**

**const example = {**

**'id': '1',**

**'subject\_id': '2',**

**'status': 'opened' as 'opened' | 'submitted' | 'graded',**

**'name': 'test subject 2',**

**'description': 'send file',**

**'deadline': '2024-05-09',**

**grade: 1,**

**}**

**Grade task**

**Description: allows teachers to grade tasks.**

**Request:**

**<?php**

**require(dirname(\_\_FILE\_\_) . '/includes/db.php');**

**$id = $\_POST['id'];**

**$grade = $\_POST['grade'];**

**// Update status to "graded"**

**$status = 'graded';**

**$sql = "UPDATE assignments SET grade='$grade', status='$status' WHERE id=$id";**

**if ($conn->query($sql) === TRUE) {**

**// Redirect to /teacher after grading successfully**

**header("Location: /teacher");**

**exit();**

**} else {**

**echo json\_encode(["error" => "Error: " . $sql . "<br>" . $conn->error]);**

**}**

**$conn->close();**

**?>**

**Response:**

**const SubmittedTasks = () => {**

**return (**

**<div>**

**{tasks.filter((task) => task.status === 'submitted').map((task) => (**

**<form action='/api/grade\_task.php' method='POST' class='task'>**

**<h3>{task.name}</h3>**

**<div class='desc'>{task.description}</div>**

**<time>{task.deadline}</time>**

**<input type='hidden' name='id' value={task.id} />**

**<input**

**name='grade'**

**min='0'**

**max='10'**

**type='number'**

**required**

**placeholder='Put your grade (0-10)'**

**/>**

**<button type='submit'>Submit</button>**

**</form>**

**))}**

**</div>**

**)**

**}**

**Submit task**

**Description: allows students to submit tasks.**

**Request:**

**<?php**

**require(dirname(\_\_FILE\_\_) . '/includes/db.php');**

**$assignment\_id = $\_POST['assignment\_id'];**

**$file\_path = 'uploads/' . basename($\_FILES['file']['name']);**

**if (move\_uploaded\_file($\_FILES['file']['tmp\_name'], $file\_path)) {**

**$sql = "UPDATE assignments SET file\_path='$file\_path', status='submitted' WHERE id=$assignment\_id";**

**if ($conn->query($sql) === TRUE) {**

**header("Location: /student");**

**echo json\_encode(["message" => "Assignment submitted successfully"]);**

**} else {**

**echo json\_encode(["error" => "Error: " . $sql . "<br>" . $conn->error]);**

**}**

**} else {**

**echo json\_encode(["error" => "Error uploading file"]);**

**}**

**Response:**

**const OpenTasks = () => {**

**return (**

**<div>**

**{tasks.filter((t) => t.status === 'opened').map((task) => (**

**<form**

**action='/api/submit\_task.php'**

**method='POST'**

**class='task'**

**enctype='multipart/form-data'**

**>**

**<h3>{task.name}</h3>**

**<div class='desc'>{task.description}</div>**

**<time>{task.deadline}</time>**

**<input name='assignment\_id' hidden value={task.id} />**

**<input type='file' name='file' />**

**<button type='submit'>Submit</button>**

**</form>**

**))}**

**</****div>**

**)**

**}**

**Submitted tasks**

**Description: allows students to see submitted tasks.**

**Response:**

**const SubmittedTasks = () => {**

**return (**

**<div>**

**{tasks.filter((task) => task.status === 'submitted').map((task) => (**

**<div class='task'>**

**<h3>{task.name}</h3>**

**<div class='desc'>{task.description}</div>**

**<time>{task.deadline}</time>**

**</div>**

**))}**

**</div>**

**)**

**}**

**Graded tasks**

**Description: allows students and teachers to see graded tasks.**

**Response:**

**const GradedTasks = () => {**

**return (**

**<div>**

**{tasks.filter((task) => task.status === 'graded').map((task) => (**

**<div class='task'>**

**<h3>{task.name}</h3>**

**<div class='desc'>{task.description}</div>**

**<time>{task.deadline}</time>**

**<div>Grade: {task.grade}</div>**

**</div>**

**))}**

**</div>**

**)**

**}**

**User manual**

The Kanban board is a tool integrated with Moodle assignments to help users manage their assignments more effectively.

**Accessing Kanban Board for students**

1. Log in to your Moodle account.
2. Navigate to the subject containing the assignments you want to manage.
3. Click on the "Kanban Board" link in the subject navigation menu to access the Kanban board interface.
4. Once you access the Kanban board, you will see a visual representation of your assignments categorized into 3 columns based on their status (e.g., "To Do," "Submitted," "Graded"). Also, you will see a button with the name of currently chosen subject.
5. Assignment card in the column “To Do” displays the title, description, due date, and contains buttons “Browse…” and “Submit” to attach and send your files. The card in the column “Submitted” shows the title, description, and the date of submission. The card in the column "Graded” displays the title, description, grade, and date of assessment.

**Accessing Kanban Board for teachers**

1. Log in to your Moodle account.
2. Navigate to the subject containing the assignments you want to manage.
3. Click on the "Kanban Board" link in the subject navigation menu to access the Kanban board interface.
4. Once you access the Kanban board, you will see a visual representation of your assignments categorized into 3 columns based on their status (e.g., "Created tasks," "Submitted," "Graded"). Also, you will see a button with the name of currently chosen subject.
5. Assignment card in the column “Created tasks” contains fields to insert the title, due date, and description of a new assignment. The card in the column “Submitted” shows the title, description, the date of submission, and the field to insert a grade together with a button “Submit”. The card in the column "Graded” displays the title, description, grade, and date of assessment.

**Creating a new assignment for teachers**

1. Choose the subject you want to add a task to.
2. Insert the title, due date, and description of a new assignment.
3. Click the button “Add task”.

**Submitting an assignment for students**

1. Choose the subject you want to submit a task to.
2. Click the button “Browse…”.
3. Choose the file you want to send.
4. Click the button “Submit”.

**Grading an assignment for teachers**

1. Choose the subject you want to grade a task of.
2. Insert a grade to the field “Put your grade (1-10). Use numbers from 1 to 10.
3. Click the button “Submit”.

The Kanban board provides a convenient interface for managing Moodle assignments efficiently. By following the instructions in this user manual, you can organize your assignments, track their progress, and ensure timely completion of tasks.