

These are the initial instructions where I will explain the expectations.

1. The goal is to populate the database with math questions from 1 to 8 grades and later up to 12 or more in such a way that each question can fit on **one or two A4 pages or be saved as a PDF document**.

Refresh

BACK

Print



The website is not designed for users to answer the questions or participate interactively.

2. Entering a question involves more than just typing text – it also includes **variables, tables, images**, and similar **HTML** elements supported by **Bootstrap**, which is pre-installed in the application as a UI foundation. Bootstrap should be used with caution because it won't render in the PDF the same way it does on screen.
3. Although the web application is responsive and can be used on a mobile device, it is recommended to use it on a computer due to better visibility. Also, PDF printing is not always possible on mobile. Bootstrap is used mainly for **tables** rather than `<div>` elements.
4. Questions are grouped by school **years – groups – and subgroups**. To start, it might be best to begin with second or third grade, as the first grade contains a lot of images and requires a more complex generation mechanism. As the grade level increases, so does the complexity of the tasks.
5. The application uses the MathJax library, which is predefined and parses the indicated code. MathJax code begins with `\(` and `\)`, and within that we use its formulas, e.g.

Solve  $\left(\frac{x}{2} + \frac{x^3}{1}\right) = \sqrt{4x} + 5$

This renders as:

$$\text{Solve } \frac{x}{2} + \frac{x^3}{1} = \sqrt{4x} + 5$$

6. The program uses variables for numbers and names. Numbers are marked with **\$\$Vx(a,b,c)\$\$**, where x is the variable index that increments, a is the starting value, b is the ending value, and c is the step. For example, **\$\$V2(50,500,10)\$\$** can generate values from the set 50, 100, 150 ... 500. The step can be a decimal (e.g., .125) but cannot be negative. The range can be negative, e.g., **\$\$V2(-5,0,.25)\$\$** will generate values in the set -5, -4.75 up to -0.25 and 0.
7. Names are marked with **\$\$Nx\$\$** – can be male or female, **\$\$Fmx\$\$** for female, and **\$\$Mmx\$\$** for male. The x is an incremented index that increases each time, regardless of gender. You cannot use **\$\$Fm1\$\$** and then **\$\$N1\$\$**, because **\$\$Fm1\$\$** will be used in both cases. Therefore, the next name must be **\$\$N2\$\$**.

An example task is available at the link, and you can view and edit it once logged in:

[ex.wi2in.com/dp/fcb4eb97-58ba-41d7-81cc-f7156995740f](https://ex.wi2in.com/dp/fcb4eb97-58ba-41d7-81cc-f7156995740f)

8. Log in for practice at [ex.wi2in.com](https://ex.wi2in.com) with the credentials:

**Username:** mi@le.na

**Password:** Matematika1!

9. On the right side, use **AddQue** and **ListQue** to edit entries. **This is a demo for presentation purposes, and you can't do anything wrong! Add, edit, delete freely!**
10. This is just an introduction to help you understand the concept. More complex and interesting elements will come later, including images and functions you'll have full control over, resulting in outputs like:

[ex.wi2in.com/dp/d274f933-3a10-4d10-985e-e1a691d40b53](https://ex.wi2in.com/dp/d274f933-3a10-4d10-985e-e1a691d40b53)

or

[ex.wi2in.com/dp/1347288a-bbcf-4240-801d-d88f2947e87c](https://ex.wi2in.com/dp/1347288a-bbcf-4240-801d-d88f2947e87c)