

# Documentation For Invoice Application

The idea of this application is to facilitate the work of one company by introducing a clear view of the invoices within the whole company. This means that each product or completed work will be recorded and it will be known exactly how much should be paid to the company. This application is "user friendly" and does not require any prior knowledge of users. It was made based on these technologies: Vue 3, Vuex and Firebase.

- **Vue3**

-Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. Vue was created by Evan You after working for Google using AngularJS in several projects. He later summed up his thought process: "I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight." The first source code commit to the project was dated July 2013, and Vue was first released the following February, in 2014. Vue 3 is the latest version of Vue. Vue 3 is basically a supercharged version of Vue 2. Vue 3 is even faster and lighter, comes with improved TypeScript support, and some great new features. The framework itself has been re-written from scratch but the API (how developers use it) remained the same.

- **Vuex**

Vuex is a state management pattern + library for Vue.js applications. It serves as a centralized store for all the components in an application, with rules ensuring that the state can only be mutated in a predictable fashion.

What is state management pattern?

- State management pattern generally use what is known as a store pattern where all actions that mutate or change the store's state are put inside a single Store class. This type of centralized state management makes it easier to understand what type of mutations could happen and how they are triggered.

Vuex is inspired by Flux, Redux and The Elm Architecture. Unlike the other patterns, Vuex is also a library implementation tailored specifically for Vue.js to take advantage of its granular reactivity system for efficient updates.

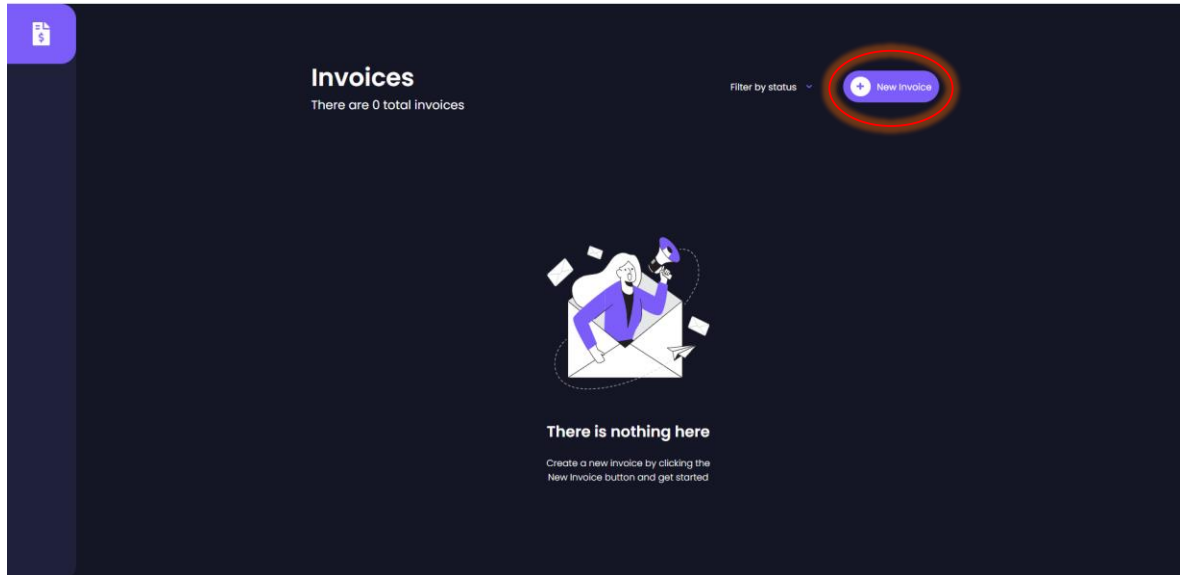
- **Firebase**

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development. Firebase evolved from Envolv, a prior startup founded by James Tamplin and Andrew Lee in 2011. They founded Firebase as a separate company in 2011 and it launched to the public in April 2012.

Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications. The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience.

- **Functionality of the Invoice app**

This is what the homepage looks like after opening the app. By clicking the new invoice button we get a form that slides from the left side of the screen.



Here we have to input the headquarters information about the firm like: Street Address, City, Zip Code and Country. And the information about the client like: Client's Name, Client's Email, Street Address, City, Zip Code and Country.

A screenshot of the 'New Invoice' form. The form is divided into two main sections: 'Bill From' and 'Bill To'. The 'Bill From' section includes fields for 'Street Address' (filled with 'Александар Македоники'), 'City' (filled with 'Македонска Каменица'), 'Zip Code' (filled with '2304'), and 'Country' (filled with 'Macedonia'). The 'Bill To' section includes fields for 'Client's Name' (filled with 'Test'), 'Client's Email' (filled with 'test@gmail.com'), 'Street Address' (filled with 'Александар Македоники'), 'City' (filled with 'Македонска Каменица'), 'Zip Code' (filled with '2304'), and 'Country' (filled with 'Macedonia'). A red circle highlights a '+ New Invoice' button in the top right. The main area features an illustration of a person in a blue suit holding a magnifying glass over a document. Below the illustration, the text reads 'nothing here' and 'invoice by clicking the utton and get started'.

By selecting the payment terms field we have 2 options, 30 days and 60 days. After choosing one of the options the payment due is automatically calculated.

The screenshot shows the invoice creation interface. At the top, there are input fields for City (Македонска Каменица), Zip Code (2304), and Country (Macedonia). Below these are fields for Invoice Date (Feb 18, 2022) and Payment Due (Mar 20, 2022). The Payment Terms dropdown is set to 'Net 30 Days'. The Product Description field contains 'test1'. The Item List section has a table with columns Item Name, Qty, Price, and Total. A red oval highlights the '+ Add New Item' button at the bottom of the table. To the right of the form, there is a 'Filter by status' dropdown and a 'New Invoice' button. An illustration of a person with a magnifying glass is shown next to the text 'nothing here' and 'Invoice by clicking the button and get started'.

| Item Name                      | Qty | Price | Total |
|--------------------------------|-----|-------|-------|
| <a href="#">+ Add New Item</a> |     |       |       |

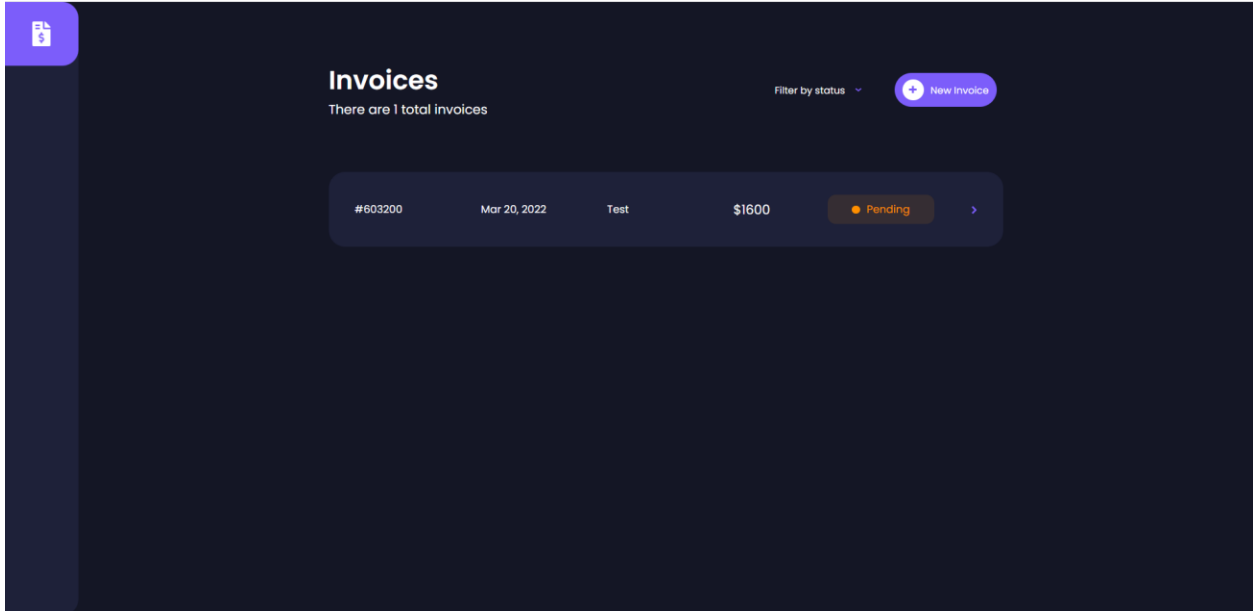
We also have the option to add items to the invoice. by clicking the Add New Item button we get another form for the requested item.

The screenshot shows the invoice creation interface after adding two items. The Item List table now contains two rows: 'item1' with Qty 20, Price 50, and Total \$1000; and 'item2' with Qty 5, Price 120, and Total \$600. The '+ Add New Item' button is still visible at the bottom of the table. The rest of the form, including the City, Zip Code, Country, Invoice Date, Payment Due, Payment Terms, and Product Description fields, remains the same as in the previous screenshot.

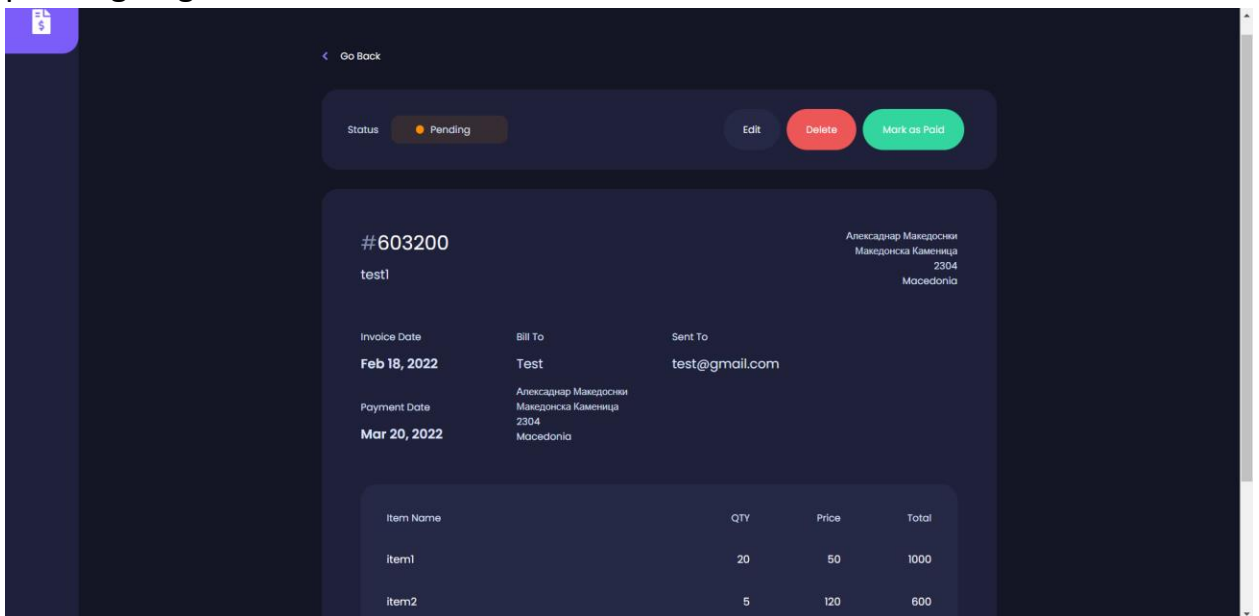
| Item Name                      | Qty | Price | Total  |
|--------------------------------|-----|-------|--------|
| item1                          | 20  | 50    | \$1000 |
| item2                          | 5   | 120   | \$600  |
| <a href="#">+ Add New Item</a> |     |       |        |

We can also delete items if we made some mistakes by making them. At the end we can create the invoice, save it as a draft or cancel it.

After creating the invoice we can see it listed on our first page. If we want to see more information about a certain invoice we can do so just by clicking on it.



By clicking on an invoice we get our second page where we can see the details about the selected invoice and do some changes if we want. We can Edit, Delete or change the status from pending to paid or from paid to pending or go back to the list of invoices.



If we choose to edit the invoice we again get the form at the left side but with already filled information.

**Edit Invoice**

**Bill From**

Street Address  
Александар Македоски

City  
Македонска Каменица

Zip Code  
2304

Country  
Macedonia

**Bill To**

Client's Name  
Test

Client's Email  
test@gmail.com

Street Address  
Александар Македоски

Buttons: Edit, Delete, Mark as Paid

Sent To  
test@gmail.com

| QTY | Price | Total |
|-----|-------|-------|
| 20  | 50    | 1000  |
| 5   | 120   | 600   |

When creating an invoice if we leave a empty text box and try to create the invoice we'll get an error message telling us to fill all the input boxes.

**New Invoice**

Filter by status ▾ New Invoice

**Bill From**

Street Address  
Александар Македоски

City  
Македонска Каменица

Zip Code  
2304

Country  
Please fill out this field.

**Bill To**

Client's Name  
Test2

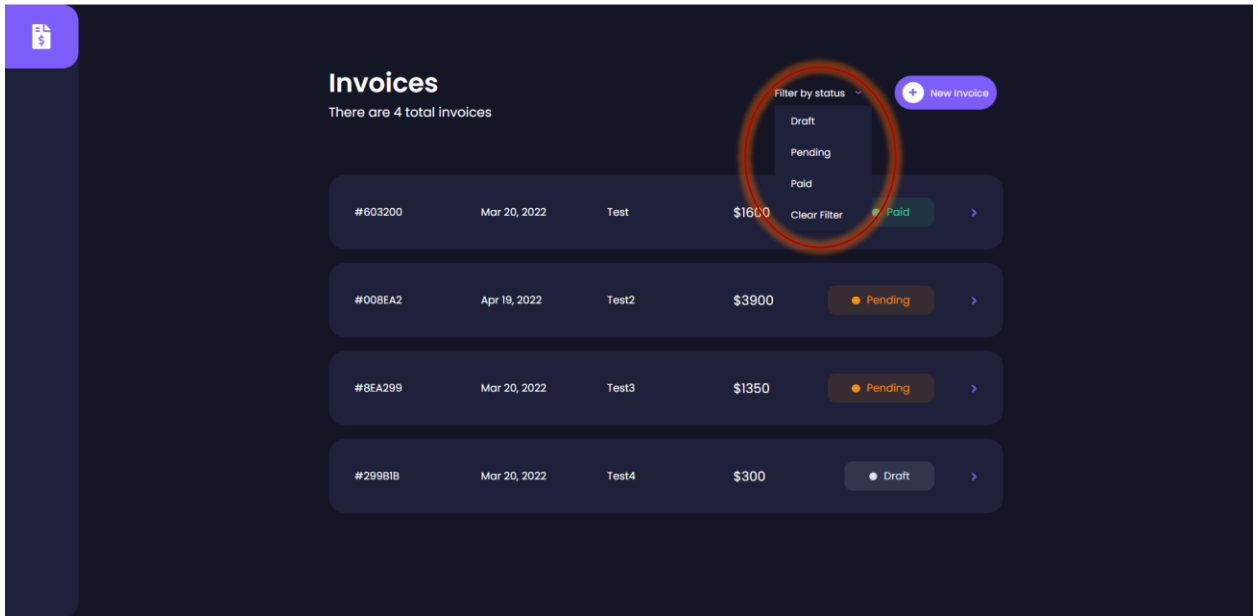
Client's Email  
test@gmail.com

Street Address  
Александар Македоски

\$1600 Paid >

| QTY | Price | Total |
|-----|-------|-------|
| 20  | 50    | 1000  |
| 5   | 120   | 600   |

There is also a filter by status for the invoices which gives us the options to filter by Paid, Pending or Draft.



Unfortunately the application does not work on Phones and is not compatible for smaller screens.

