WebSocket and payment integration

WebSocket/Real-time Features

Socket.IO Implementation

Server Setup (in `index.js`):

```
const server = http.createServer(app);
const io = socketIo(server, {
  cors: { origin: "*" }
});
handleChat(io);
```

Chat System

Features:

- Real-time messaging between buyers and sellers
- Chat room management
- Online status
- Typing indicators
- Unread message count

Events:

- `connection` - Client connected
- `join_room` - Join specific chat room
- `send_message` - Send message
- `typing` - User typing indicator
- `read_messages` - Mark messages as read
- `disconnect` - Client disconnected

Implementation (`services/chat.services.js`):

```
io.on('connection', (socket) => {
  socket.on('join_room', (roomId) => {
    socket.join(roomId);
  });

  socket.on('send_message', async (data) => {
    // Save message to MongoDB
    // Emit to room participants
    io.to(roomId).emit('new_message', message);
  });
});
```

Payment Integration

Xendit Integration

1. Create Virtual Account

```
const { xenditClient } = require('./config/api');

// Create VA
const response = await xenditClient.VirtualAccount.create({
  external_id: orderNumber,
  bank_code: 'BCA',
  name: customerName,
  expected_amount: totalAmount,
  expiration_date: expirationDate
});
```

2. Webhook Handling

```
// POST /webhook/xendit
app.post('/webhook/xendit', async (req, res) => {
  // Verify webhook signature
```

```
  const signature = req.headers['x-callback-token'];
  if (signature !== process.env.XENDIT_CALLBACK_TOKEN) {
    return res.status(401).json({ error: 'Unauthorized' });
  }

  // Process payment
  const { external_id, status } = req.body;
  if (status === 'PAID') {
    await updateOrderStatus(external_id, 'PAID');
  }

  res.json({ success: true });
});
```

BRI Virtual Account (BRIVA)

Additional integration for BRI-specific virtual accounts in `services/briva/`.

Revision #1
Created 24 February 2026 09:28:04 by ondeliveloper
Updated 24 February 2026 09:29:02 by ondeliveloper